PDQ-3

Q-BUS & UCSD PASCAL: THE COMPREHENSIVE PACKAGE from ADVANCED COMPUTER DESIGN

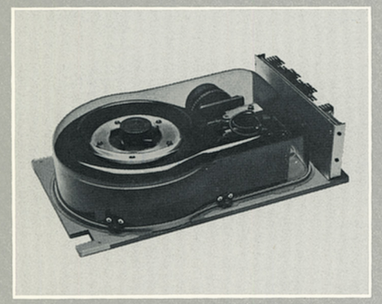
PDQ-3

A powerful, low-cost system that combines state-of-the-art technology with solid field performance. PDQ-3's unique expansion capability allows it to grow with user needs while continuing to provide the highest level of performance and flexibility.

•Solid Hardware Reliability
•Low Software Maintenance
•Complete Installation & Service

LSI-11 'Q'-BUS

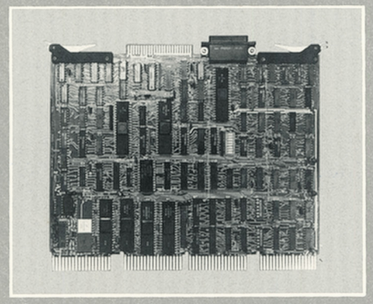
PDQ-3 incorporates the industry *de facto* standard LSI-11 "Q"-Bus from Digital Equipment Corporation (DEC). The Q-Bus permits the addition of a wide range of peripherals including disk and magnetic tape controllers, A/D, D/A, digital I/O, communication



multiplexers and interfaces, and graphic interfaces. The unique flexibility of the Q-Bus, combined with the power of Pascal, makes PDQ-3 the most expandable, cost-effective multiprocessing system in its price range.

COMPACT-VERSATILE

The PDQ-3 CPU module features a high performance 16-bit stack-based processor, real-time clock, interval timer, multiplexed CRT and serial printer interface, and bootstrap prom, all packaged on a 8½″ × 10″ (quad size) multilayer printed circuit board. Its compact



design combines engineering excellence with space-saving economy, providing room for future expansion within the backplane.

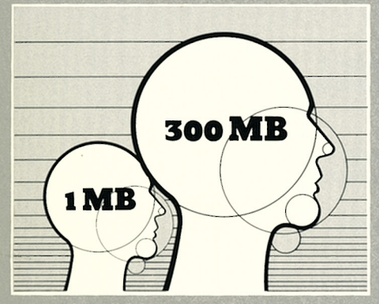
UCSD PASCAL

PDQ-3 features direct execution of UCSD Pascal Version III.0, permitting a high degree of software portability while dramatically reducing software development and maintenance costs. An exclusive package of software enhancements devel-

oped by ACD incorporates such new Pascal constructs as Process, Wait, Signal, and Start; the ACD software system provides for a multiprocessing operating system with a real-time interrupt-driven I/O system. Other enhancements found exclusively on PDQ-3 include a spooler, advance system editor, and command file processor. The UCSD Pascal system also contains a file handler, linker, and Pascal compiler.

EXPANDABLE to 300 MB

Because of its bus structure, PDQ-3 can interface from 1 MB to 300 MB of disk storage utilizing Control Data Corporation SMD disk drives with controllers. Also available are Winchester technology and cartridge removable disk drives, as well as internally



mounted 8″ Winchester disk drives with up to 24 MB capability.

A POWERFUL LOW-COST SYSTEM

For the $6,995 selling price, PDQ-3 includes: PDQ-3 processor, 64 KB of MOS memory, real time clock, interval timer, CRT console and printer interface, DMA floppy disk controller, two (2) 8″ single density floppy disk drives, four (4) slot 11/23, Q-Bus backplane with

$6,995

a 1.67 MB throughput rate, chassis, power supply and the complete UCSD Pascal software system. Complete installation and after-sale service is available from Associated Computer Engineers, an affiliate of Advanced Computer Design.

Specifications

The PDQ-3 CPU module integrates all of the features listed below on a single, multilayered 8½" × 10" LSI-11* compatible quad board.

Features:

- High performance, 16-bit stack-oriented processor microcoded for direct execution of UCSD Pascal** "P" Code Version III.0.
- Execution times up to ten (10) times faster than comparable systems.
- Directly addressing to 128 KB of MOS memory.
- LSI-11 bus compatible peripheral controllers and memory modules.
- UCSD certified operating system with job control commands, UCSD Pascal compiler, file manager, screen oriented editor and support utilities.
- Fully interrupt driven with multitasking and concurrency primitives.
- Supports multiprocessing environment - one root program can create multiple, prioritized tasks to control a multiterminal application.
- Extensive instruction set for Stack, Array, Set, Block Move, Byte, Logical, Real, Integer, Signal and Wait Processing.
- Floating point hardware (IEEE standard).
- Highspeed 16-bit integer arithmetic with hardware multiply and divide.
- Programmable real-time clock and interval timer.
- DMA floppy disk controller for up to four (4) drives with double or single density selection and disk formatting under software control.
- Bootstrap loader for floppy disk.
- Powerful hexadecimal programmer's console debugger.
- RS232-C serial interface with eight (8) switch selectable baud rates up to 19,200 baud.
- Power fail/auto restart, with provision for battery back-up.

*Trademark of Digital Equipment Corporation
**Trademark of Regents of University of California

Advanced Computer Design

Advanced Computer Design
7584 Trade Street
San Diego, CA 92121
(714) 578-9595

Advanced Computer Design

A Subsidiary of Computer Services Group, Inc. / 7584 Trade St. / San Diego, CA 92121 / (714) 578-9620

I thought that I would take this opportunity to send you a documentation on some of the latest features and capabilities available on the PDQ-3; for concurrency and multi-tasking, for its operating system and utilities, and for its new Advanced System Editor which increases programmer productivity by 30% to 40%. I'm also enclosing the table of contents and first chapter of some of the PDQ-3 Reference Manuals.

After reading the enclosed materials, I'm sure that you will agree that the PDQ-3 with its direct, high speed execution of high level language statements, large direct memory addressing space, and the expansion capabilities of the DEC LSI-11 "Q" Bus, which allows you to plug in hundreds of different interfaces from magnetic tape and disc controllers to graphics and communications subsystems, makes for a very cost effective, expandable system for your current and future computer needs.

Our customer have been very impressed with the performance capabilities of the PDQ-3. The PDQ-3 is considered the best UCSD "P" machine currently available. We have had several users running large, complex applications that they could previously only run on DEC's large scale 32 bit VAX 11/780 systems. With all of its power and expansion capabilities, the PDQ-3 is quite a value when you consider that it offers up to 10 times the performance of an APPLE III at around the same price.

If you should have any questions on the PDQ-3 or would like to set up a meeting to see the PDQ-3 and the operational and applications software available for it, please feel free to give me a call at 714/578-9620. Quantity/ OEM discounts are available.

Sincerely yours,

Gary Noonan
Sales Manager

Enclosures

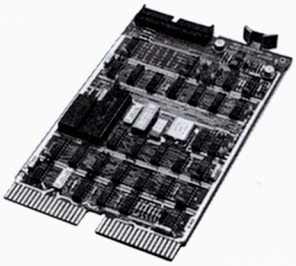
MDB INTERFACE PRODUCTS for use with Digital LSI*-11/2 & 11/23 Computers PRODUCT DESCRIPTION

Except where noted below in the individual product descriptions, all modules described below are compatible with DEC supported Operating Systems, software and diagnostics.

LSI-11 COMMUNICATIONS/TERMINAL INTERFACE MODULES

Asynchronous Serial Interfaces

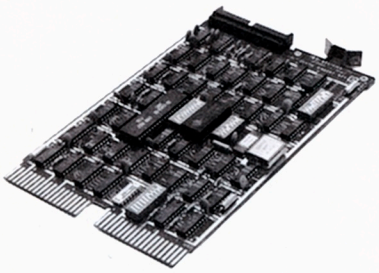
MLSI-DLV11



Single line EIA RS-232-C/20 ma Current Loop serial interface. Switch selectable: device addressing, interrupt vectors, UART parameters and baud rates from **50** to **19.2K baud.** Generates Reader-Run for ASR type terminals. Active or passive receiver and/or transmitter current loop circuitry. Utilizes optional BCO5M-2C (current loop) or BCO5C-25 (RS-232) cables. (See Note 1 on last page)

Mounting Code	Power Drawn	Bus Loads Drawn
Dual slot	.8a @ +5V .04a @ +12V	1 max

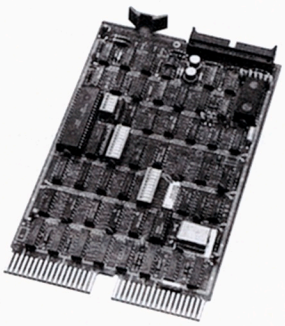
MLSI-DLV11-E



Single line EIA RS-232-C serial interface with modem control. Switch selectable: device addressing, interrupt vectors, UART parameters and baud rates from **50** to **19.2K baud.** EIA RS-232-C drivers and receivers for complete dataset control. Split transmit and receive baud rate capability. Boot or Halt on framing error is user strap selectable. Provisions for user supplied external clock inputs for baud rate control. Utilizes optional BCO5C-25 cable. (See Note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Dual slot	1a @ +5V .1a @ +12V	1 max

MLSI-DLV11-F



Single line EIA RS-232-C/20 ma Current Loop serial interface with programmable **and** switch selectable baud rates from **50** to **19.2K baud. Four level interrupt.** Switch selectable device addressing, interrupt vectors and UART parameters. Generates Reader-Run for ASR type terminals. User strap selectable Boot or Halt on framing error. UART cleared by user selectable INIT or BDCOK signal. Program control of maintenance bit; loop back cable not required. Also includes a **Buffer Ready/Printer Busy monitor circuit** to allow RS-232 data transmission to printers or other similar devices without buffer overrun. (See Note 2 on last page) Utilizes optional BCO5M-2C or BCO5C-25 cables. Backwards compatible with MDB and DEC DLV11. (See Note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Dual slot	.8a @ +5V	1 max

MDB SYSTEMS INC.
1995 N. Batavia Street
Orange, California 92665
714-998-6900
TWX: 910-593-1339

Date: March 1, 1981

*TM Digital Equipment Corporation

MLSI-DLV11-FX

Single line serial interface with combined EIA RS-232-C/20 ma Current Loop and RS-422 circuitry on a single board. Switch and programmable baud rates from **50** to **19.2K baud.** Switch selectable device addressing, interrupt vectors and UART parameters. Generates Reader-Run signal. Also includes **Buffer Ready/Printer Busy** user strap selectable circuitry. (See Note 2). Utilizes optional BCO5M-2C or BCO5C-25 cables. RS-422 cable supplied to user requirements. Contact factory for price and delivery of RS-422 cables. (See Note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Dual Slot	.8a @ +5V	1 max

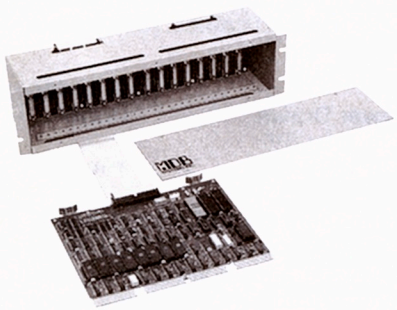
MLSI-DLV11-J

Four port asynchronous EIA serial interface.

Mounting Code	Power Drawn	Bus Loads Drawn
Dual Slot	1.0a @ +5V 0.25a @ +12V	1 max

Asynchronous Multiplexors

MLSI-DZ11-A

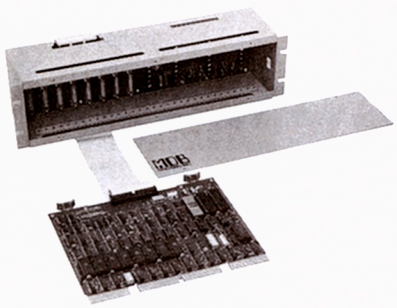


Asynchronous 8-line EIA multiplexor which provides all features of the DEC Unibus* DZ-11-A plus each line has programmable character formats and data rates from **50** to **19.2K baud.** Switch selectable device addressing and interrupt vectors. Includes dataset control. **Four level interrupt.** Multiplexor contains a 64 character buffer with a 16 character SILO counter to allow minimal processor intervention, which results in greater throughput. Compatible with DEC operational and diagnostic software. Comes with MDB-H317-E, 16 channel distribution panel, test connectors and cabling between MLSI-DZ11 quad module and distribution panel. (See Note 1) Upgradable to a 16 line multiplexor by use of MLSI-DZ11-B.

Mounting Code	Power Drawn	Bus Loads Drawn
Quad slot and Distribution Panel	2.2a @ +5V .1a @ +12V	1 max

(**Note:** The DEC DZV11 multiplexor is a four port unit and requires that the cable plug directly onto the module. It does not utilize a communications panel.)

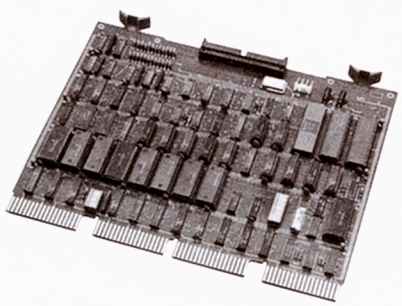
MLSI-DZ11-AC



Asynchronous 8-line multiplexor with combined EIA/20 ma capability on a per line basis (provides operational features of DEC Unibus* DZ11-A and DZ11-C within a single board). Each line is programmable from **50** to **19.2K baud.** Switch selectable device addressing and interrupt vectors. Includes dataset control. **Four level interrupt.** Multiplexor contains a 64 character buffer with a 16 character SILO counter to allow minimal processor intervention, which results in greater throughput. Compatible with DEC operational and diagnostic software. Includes MDB-H317-AC, 8 channel EIA/20 ma distribution panel, test connectors and cabling between MLSI-DZ11 quad module and distribution panel. (See note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Quad slot and Distribution Panel	2.2a @ +5V .1a @ +12V	1 max

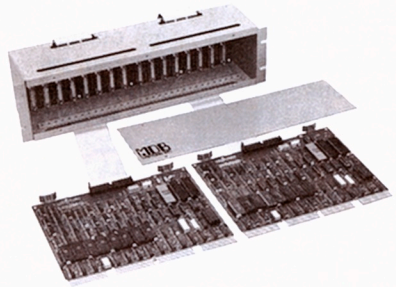
MLSI-DZ11-B



Asynchronous 8-line EIA multiplexor module which provides all features of the DEC Unibus* DZ11-B plus each line has programmable character formats and data rates from **50** to **19.2K baud.** Switch selectable device addressing and interrupt vectors. Includes dataset control. **Four level interrupt.** Multiplexor contains a 64 character buffer with a 16 character SILO counter to allow minimal processor intervention, which results in greater throughput. Comes with cable to plug into MDB-H317-E distribution panel and test connector. Used to upgrade an MLSI-DZ11-A 8-line multiplexor to an MLSI-DZ11-E 16-line multiplexor. (See Note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Quad slot	2.2a @ +5V .1a @ +12V	1 max

MLSI-DZ11-E



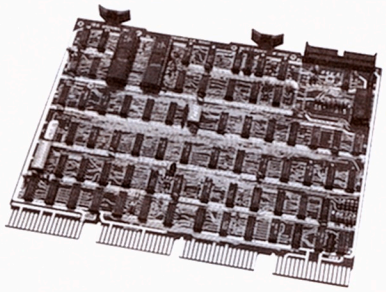
Asynchronous 16-line multiplexor which provides all features of DEC Unibus* DZ11-E plus individually programmable character formats and data rates for each EIA line from **50** to **19.2K baud.** Switch selectable device addressing and interrupt vectors. **Four level interrupt.** Multiplexor contains a 64 character buffer with a 16 character SILO counter to allow minimal processor intervention, which results in greater throughput. Includes dataset control. Comes with MDB-H317-E, 16 channel distribution panel, test connectors and cabling between 2 MLSI-DZ11 quad modules and distribution panel. (See note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
2 Quad slots and Distribution Panel	4.4a @ +5V .2a @ +12V	2 max

Note: The MDB Distribution Panels measure 19" wide x 5¼" high x 4½" deep, and fit into a standard RETMA rack.

Single Line Synchronous Interfaces

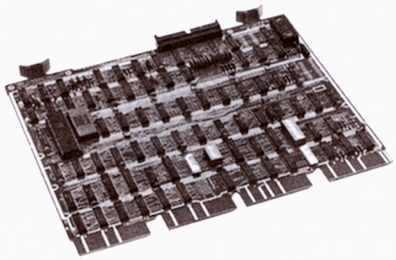
MLSI-DUV11



Single line Synchronous serial interface. Complete modem control. Synchronous or Isochronous communications modes, either half or full duplex. Program selectable USART parameters. DIP switch selectable device addressing and interrupt vectors. Transmitter and receiver double buffered logic permits a full character time for handling of interrupts. Three program controlled maintenance modes; System Test, Internal Loop Test, External Loop Test. Provides level conversion between on-board TTL levels and EIA RS-232 or appropriate dataset levels and accommodates **transmission rates up to 40K baud.** Operates with Bell 303/313-type or 201/202-type datasets. Includes BCO5C-25 cable. (See Note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Quad slot	1.5a @ +5V .1a @ +12V	1 max

MLSI-DUPV11



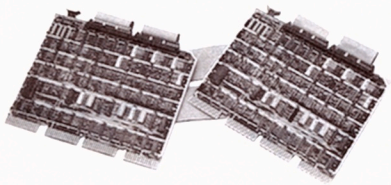
Single line Synchronous serial interface. Complete modem control for full or half duplex operation. Provides all features of DEC Unibus DUP11-DA, including on board hardware **CRC** or **LRC** checking and generation for **B**it **O**riented **P**rotocols **SDLC, ADCCP** and **HDLC.** Also **B**yte **C**ontrol **P**rotocols BI-SYNC and DDCMP. **X.25 capability.** Programmable character length 1 to 8 data bits for BOP or 5 to 8 bits for BCP. **Data rates to 500K bps. Four level interrupt.** Error control CRC-CCITT pre-set of 0's or 1's, CRC-16 preset to 0's, VRC odd or even or no error control. Data to or from the module is double buffered for full character service time. Incorporates a Multi-Protocol Communications Controller (MPCC) chip. Provides level conversion between on-board TTL levels and EIA or dataset levels. Includes BCO5C-25 cable. (See Note 1)

Mounting Code	Power Drawn	Bus Loads Drawn
Quad slot	2a @ +5V .2a @ +12V	1 max

NOTE: As used herein, selectable UART or USART parameters encompass selection of 5, 6, 7 or 8 bit characters; odd, even or no parity, and 1, 1.5 or 2 stop bits

LSI-11 INTERPROCESSOR LINK SUBSYSTEMS

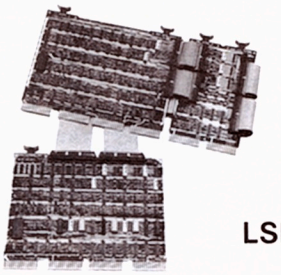
MLSI-DA11BOI



Optically isolated parallel DMA interprocessor link between any two LSI-11 computers. High speed differential drivers coupled with optically isolated receivers maximize circuit isolation and provide ground loop current elimination for bidirectional **data transfer rates up to 500K words per second.** Data rate adjustments provided for optimum system operation over distances up to 1,000 feet. Selectable feature allows data transfers across 32K boundaries in blocks of up to 32K words. Switch selectable address and interrupt vector. Adjustable data transfer rates for regulating DMA load on each computer. Compatible with DEC DR11-B and DA11-B operating and diagnostic software. (See Note 1). Cabling provided for operation between two computers up to 50 feet, optional lengths up to 1,000 feet.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
		1 max
Quad slot (each computer)	2.6 a (each computer)	(each computer)

MDB/MLSI-DA11BOI



Optically isolated parallel DMA interprocessor link between any LSI-11 computer and any Unibus computer. (PDP-11 or VAX*) All features are identical to the MLSI-DA11BOI.

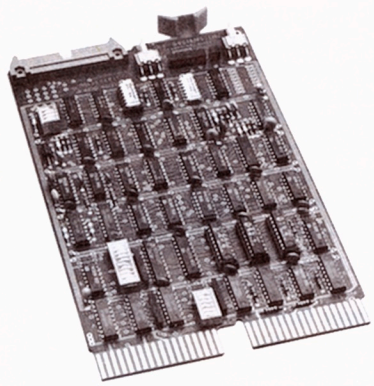
Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
		1 max
Quad slot in LSI-11 backplane Hex slot in Unibus backplane	2.6a (each computer)	(each computer)

LSI-11 PERIPHERAL DEVICE CONTROLLER MODULES

(Parallel Data Transfer)

(Customer to specify peripheral manufacturer and its model number.)

MLSI-LP11



Line printer controller for all popular line printers. Interfaces include **Centronics, Dataproducts, LA 180, G.E. TermiNet**,** and **Houston Instrument.** Also operates printers emulating Centronics or Dataproducts interfaces such as **Printronix, Mannesmann/Tally, Okidata, CDC** etc. **Four level interrupt.** Compatible with DEC LP11 or LS11 diagnostics and printer driver routines. (See Note 1) **Has MDB exclusive PrinTest (TM) and Loopback features, plus LEDs to give visual indication of data lines.** (Note 3) PrinTest feature has the capability of being operated remotely from the printer if a low-going signal is entered on a prescribed pin of the printer interface connector. DIP switch selectable addressing and interrupt vectors. Furnished with 15-foot cable which contains the proper interface connector for the printer. Long line capability supplied by separate Universal Long Line Adapter Box (MDB-ULLAB-03) to allow placement of printer up to 3,000 feet from the computer.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.7 a	1 max

MLSI-LP11-A

Line printer controller with Data Printer interface. Contains all features of LP11 described above; however, interface is designed to operate exclusively with most models of printers manufactured by Data Printer. The printer must have the standard interface, not the first character interface. Supplied with 15-foot cable. Use ULLAB for Long Line operation.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.8a	1 max

MLSI-LV11

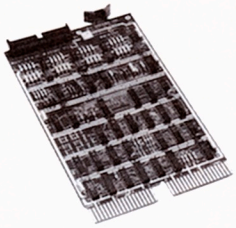
High speed electrostatic printer/plotter controller for Versatec or similar emulating device. Compatible with DEC **LV11** operating and diagnostic software. **Not** compatible with Versatec supplied software. (See Note 1). Includes 15-foot cable standard.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.8a	1 max

(TM) MDB Systems

**TM General Electric Co.

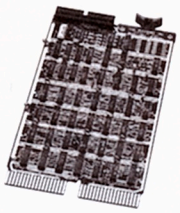
MLSI-XYV11



Parallel incremental plotter controller for **Houston Instrument** or **CalComp** XY plotters or equivalent. Multiple plotter address selection and differential drivers standard. Compatible with DEC **XY-11** operating and diagnostic software. (See Note 1). Includes 15-foot cable that **does not** include plotter mating connector.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.9a	1 max

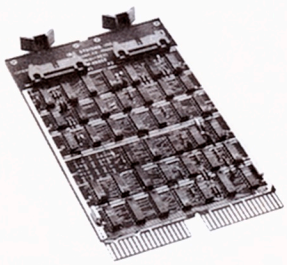
MLSI-CR11



Card reader controller for all speed versions of **Documation, Truedata** and other popular card readers. Multiple card reader address selection standard. Compatible with DEC **CR11** operating and diagnostic software. (See Note 1). 15-foot cable with mating connector standard.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.9a	1 max

MLSI-PC11



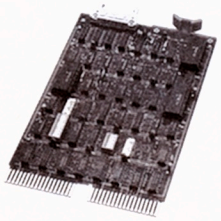
High speed paper tape reader/punch controller for popular paper tape reader/punch devices. Interfaces for **Remex, Digitronics, EECO** and other popular makes. **Not compatible with DEC manufactured paper tape Reader/Punches.** Compatible with DEC **PC-11** operating and diagnostic software. (See Note 1). 15-foot cable with mating connectors standard.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.9a	1 max

LSI-11 SYSTEM MODULES

(Parallel Data Transfer)

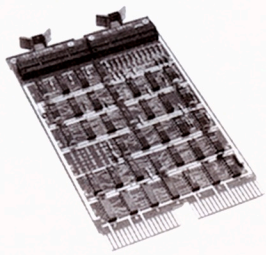
MLSI-IBV11



IEEE/488 instrumentation bus controller. Provides interface between LSI-11 computer and programmable instruments that conform to ANSI/IEEE std. 488-1975. **Four level interrupt.** Operating and programming considerations are exactly as described for DEC's **IBV11-A.** Includes 12-foot cable with connectors for first instrument.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.8a	1 max

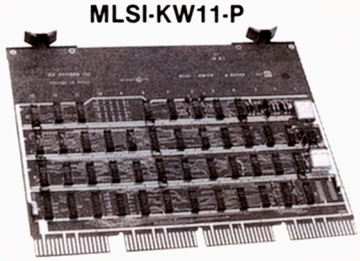
MLSI-DRV11-C



Parallel line interface module with multiple address selection and multiple interrupt vectors. Provides 16-bit word programmed data transfers with four control lines to peripheral units. **Up to 90K words/second transfer rate.** Contains an Output buffer register, Input buffer register, and control and status register, has two 20 pin Berg type connectors. Optional cables available. Compatible to DEC **DRV11** operating and diagnostic software.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	1.3a	1 max

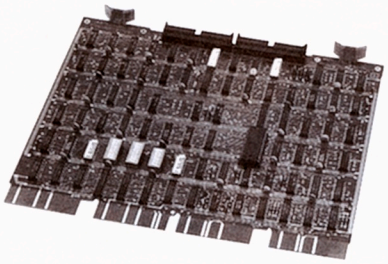
MLSI-KW11-P



Programmable real time clock. Provides four clock rates, program selectable (external, 60 Hz, 10KHz, 100KHz). Has own 60Hz crystal controlled oscillator. Crystal control is with external inputs in three modes of operation (single interrupt, repeat interrupt, external event). Compatible with DEC **KW11-P** operating and diagnostic software. (See Note 1).

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Quad slot	1a	1 max

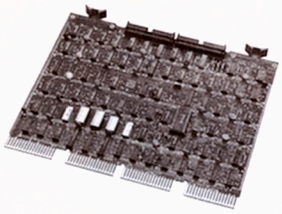
MLSI-DR11-B



Direct Memory Access module compatible with DEC Unibus operating and diagnostic software. (See Note 1). Provides interrupt request and bus master control for bi-directional exchange of 16-bit data from the LSI-11 bus to an external device. Strappable for compatibility to DEC DRV11-B operating and diagnostic software. Electrically compatible with DEC DRV11-B with strap changes. Four registers, consisting of an input and output buffer (16-bit read/write), control status (16-bit read/write), form the interface logic circuits of the DR11-B module. Additional user selectable logic allows data transfer across 32K boundaries without error condition (overflow to address zero will cause ERROR). **I/O Plug compatible with MDB-DR11B for Unibus computers.**

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Quad slot	2.5a	1 max

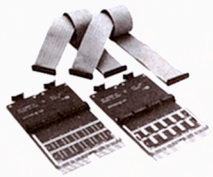
MLSI-DRV11-B



Direct Memory Access module compatible with DEC operating and diagnostic software. Provides interrupt request and bus master control for bi-directional exchange of 16-bit data from the LSI-11 bus to an external device. **I/O Plug compatible** with DEC DRV11-B LSI-11 system module.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Quad slot	1.8a	1

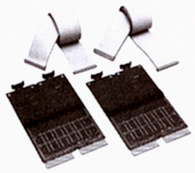
MLSI-BCV-000



Backplane jumper/cable assembly for use between first and second chassis in multiple chassis configuration. 250 ohm termination in first system; second should be terminated separately. Cable length 6 feet.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot in each chassis	.7a	N/A

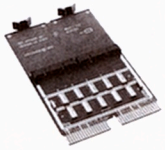
MLSI-BCV-001



Backplane jumper/cable assembly for use between second and third chassis in multiple chassis configuration. Not terminated on either end. Cable length 4 feet.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot in each chassis	N/A	N/A

MLSI-TEV-003



Bus terminator module with 120 ohms termination.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	1.4a	N/A

MLSI-REV-11A

Bootstrap refresh/termination module. (Not for use with 11/23 systems)

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual Slot	1a	1 max

MLSI-REV-11C

Bootstrap refresh option.

Mounting code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	1 a	1 max

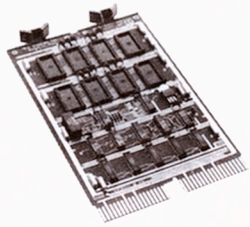
MLSI-BGC

LSI-11 bus grant continuity card.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
One-half dual slot	N/A	N/A

GENERAL PURPOSE LSI-11 (P)ROM MEMORY MODULES

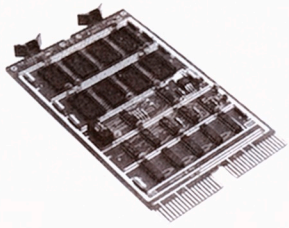
MLSI-MRV-000



PROM module with 8 sockets accommodating commercially available **2704, 2708** or **3624 PROMS** or equivalent. Switch selectable memory area allocation. Capacity from **512 to 4K 16-bit words.** On board regulated -5VDC supply.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.4a (plus 70ma-12V)	1 max

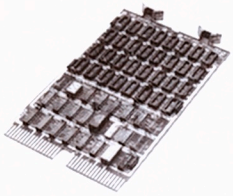
MLSI-MRV-001



PROM module with 8 sockets accommodating **1702 PROMS** or equivalent. Switch selectable memory area allocation. Capacity from **256 to 1K 16-bit words.** On board regulated -9VDC supply.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.4a (plus 70ma-12V)	1 max

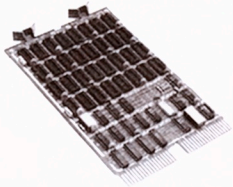
MLSI-MRV-002



PROM module with 32 sockets accommodating commercially available **5623, 5624 or equivalent PROMS.** Switch selectable memory area allocation. Capacity from **256 to 4K 16-bit words.**

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.4a	1 max

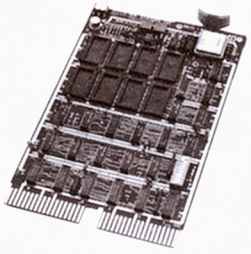
MLSI-MRV-003



PROM module with 32 sockets accommodating commercially available **3625 or equivalent PROMS.** Switch selectable memory area allocation. Capacity from **256** to **4K 16-bit words.**

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.4a	1 max

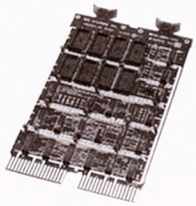
MLSI-MRV-004



PROM module containing 8 sockets with **on-board programmer.** Accommodates commercially available single voltage **2716, 2732 and 2758 or equivalent PROMS.** Programmer and memory area allocation are both switch selectable. Programming of PROMS accomplished by doing memory to memory transfer with 55 msec. wait loop. Capacity from **1024 to 16K 16-bit words.** Expansion in 1K, 2K or 4K increments. On-board +25VDC supply.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.5a (+.02a of +12V)	1 max

MLSI-MRV-005

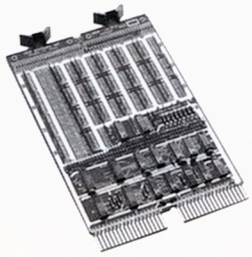


PROM/RAM combination module with 8 sockets. Accommodates **2716, 2732, 2758** or equivalent type PROMS as well as **4118 RAMS.** (Chip addressing must match). Capacity from **1K** to **4K** words for 4118; 2716 gives up to **8K** in **2K** increments; 2758 provides **8K** in **1K** increments and 2732 gives up to **16K** in **4K** increments.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.4a (plus PROM current)	1 max

LSI-11 GENERAL PURPOSE INTERFACE AND FOUNDATION MODULES

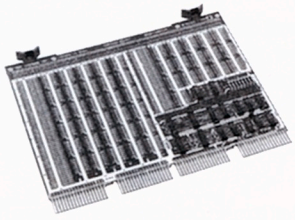
MLSI-1710



General purpose bus foundation module. Basic Q-bus programmed I/O interface, address decode circuitry and multiple interrupt vector priority selection. Multiple external I/O provisions on module. Pinned for **20 wirewrap IC positions** for user designed logic. Wirewrap pins are on component side of module. Will accommodate any .300", .400" or .600" center DIP packages. Provision for variety of cable connector positions.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	.4a	1 max

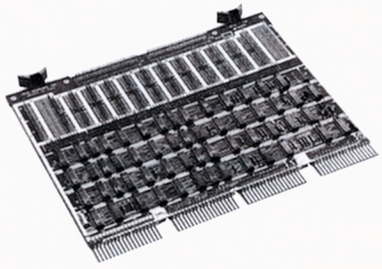
MLSI-DRV11-P



General purpose bus foundation module. Same as MLSI-1710 but has available up to **56 IC positions** for user designed logic. Provision for wide variety of cable connector positions.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Quad slot	.4a	1 max

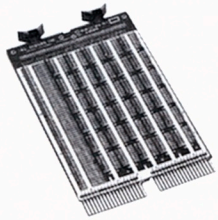
MLSI-11B



General purpose Direct Memory Access bus foundation module. Contains basic Q-bus programmed I/O interface with address decode circuitry, interrupt vector priority selection, plus address word count, data registers and bus master control and timing logic. Includes **26 IC positions** for user designed logic and multiple external I/O provisions. Wirewrap pins are on component side of module. Will accommodate any .300", .400" or .600" center DIP packages. Provision for two cable connector positions, up to 50 pins each.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Quad slot	1.6a	1 max

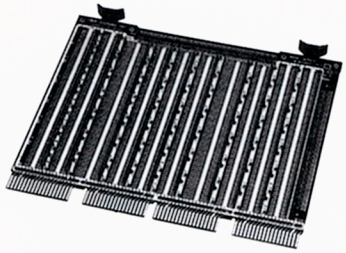
MLSI-WWB1



Universal dual wirewrap module. Wirewrap pins provided for **36 IC positions** on component side of module. Has multiple external I/O provisions. Will accommodate any .300", .400" or .600" center DIP packages. Provisions for 4 cable connector positions, up to 50 pins each.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	User Circuit Dependent	N/A

MDB-W9501



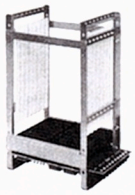
Universal quad wirewrap module. Accommodates IC sockets from 14 to 40 pins, in any combination and location. Power and ground decoupling pads provided. Wirewrap pins mounted on component side of module. Provision for wide variety of cable connector positions in two different locations.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Quad slot	User Circuit Dependent	N/A

NOTE: Dimensions of wirewrap pins are 0.025" square, length is 0.318".

LSI-11 SYSTEM CHASSIS AND ENCLOSURE SUBASSEMBLIES

MLSI-BPA82



Backplane/Card Guide assembly providing 8 standard LSI-11 Q-bus dual slots: accepts one LSI-11-2 or LSI-11/23 processor and seven dual modules. **Compact assembly fits easily into 5¼" high RETMA chassis.** Provides removable pluggable 120 ohm bus termination. Terminal posts for standard LSI-11 and custom power requirements. Prewired printed circuit LSI-11 backplane. Full length card guides for maximum board support. Compact 10.50" x 4.75" x 7.30" assembly.

MLSI-BPA84

Backplane/Card Guide assembly providing **8 quad or 16 dual** standard LSI-11 Q-bus slots. Compact assembly fits easily into 5¼" RETMA chassis. Standard LSI-11 bus accepts one LSI-11/2 or LSI-11/23 processor and 15 dual modules. Also accommodates older LSI-11 (KD11-F) processor. Terminal posts for standard LSI-11 and custom power requirements. Full length card guides for maximum board support. Pre-wired multilayer printed circuit LSI-11 backplane. Compact 10.50" x 4.75" x 12.58" assembly. Used in some versions of MDB System Boxes for LSI-11 computers and in LSI-11 Expansion Boxes.

MLSI-BPA84-A

Backplane/Card Guide assembly identical to BPA84 above except for **two dual slots, 6 C-D** and **7 C-D.** These two slots are uniquely wired to accept DEC's RLV11 or RLV21 disk controller. All other slots are wired for Q-bus. Used in some versions of MDB System Boxes and Expansion Chassis.
See page 14 for backplane configurations.

MLSI-H984

See page 13 for Photograph

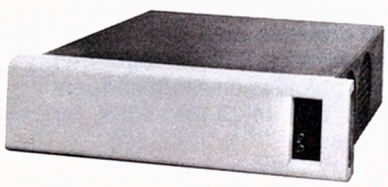
Roll-around system cabinet, outside dimensions: 30" Height, (includes ball casters), Depth, 31", Width 21". Inside dimensions: 24.5" Height, Depth 29", Width 19". Weight 80 pounds. Rear panel with cable cut-out, louvered side and rear panels. Internally mounted A/C power strip. External on/off circuit breaker. Two sets of vertical mounting rails, RETMA spacing. One pair, 19" long horizontal mounts. Walnut grain formica top, baked enamel light gray (No. 26622 FED STD 595) finish.

MLSI-H961-A

See Page 13 for Photograph

High-boy system cabinet with 60" panel height. Removable side and back panels. Rear panel is hinged for easy opening. Top mounted cooling fan and filter (590 cfm air circulation). Internally mounted AC power strip. Equipped with front stabilizer feet with leveling pads, kickplate and casters. Inside dimensions (RETMA standard): Height 63", Depth 25", Width 19". Outside dimensions: Height 69.75", Depth 30", Width 21.75". Weight 240 pounds. Finish is baked enamel, light gray (No. 26622 FED STD 595) with black trim.

MLSI-BA11-20X*

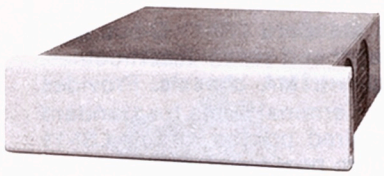


System chassis assembly for LSI-11 processors and associated modules. Includes front panel, front loading **BPA84** backplane/card guide assembly and triple output regulated switching power supply, with +5v @ 25a, +12v @ 5a and −12v @ 1 a. Two cooling fans for 200 cfm air circulation. Includes bus terminator, power on/off failure sequencing and Line Time Clock. Front panel switches and indicators for user selected DEC LTC or KW11-L compatibility. Front panel indicators: RUN and DC On. Front panel switches: RUN, LTC enable/disable and HALT. The AC On/Off power switch is accessable by simply pulling off the front panel. Can accommodate 16 dual size boards or 8 quad boards. The last dual slot is occupied by a combination bus terminator and LTC module. Unit also includes pseudo switch register for PDP-11* compatibility. All cables exit from rear through an adjustable strain relief cable slot. RETMA rack mountable. Dimensions: 5¼" high, 19" wide and 15" deep. Weight: 30 pounds.

MSLI-BA11-21X*

System chassis assembly same as MLSI-BA11-20X except includes the **BPA84-A** backplane/card guide assembly for use with RL01 or RL02 disk controllers. Last dual slot in backplane is occupied by bus terminator and LTC module functions. Dimensions, power and weight of RETMA rack mountable chassis assembly are the same as unit above.

MLSI-BA11-30X*

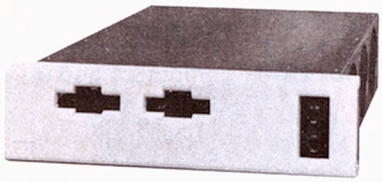


Expansion chassis assembly with blank front panel, **BPA84** backplane/card guide assembly. Triple output regulated switching power supply with +5v @ 25a, +12v @ 5a and −12v @ 1a. Front loading chassis holds 8 quad or 16 dual size modules. Two cooling fans for 200 cfm air circulation. Strain relief for rear exiting cables. Dimensions: 5¼" high, 19" wide, 15" deep. RETMA rack mountable. Weight: 30 pounds. Use MLSI-BCV-000 for connection to CPU chassis assembly.

MLSI-BA11-31X*

Expansion chassis assembly same as MLSI-BA11-30X above, but, contains **BPA84-A** backplane/card guide assembly for use with RL01 or RL02 disk controllers. Weight, dimensions and power same as unit above.

MSLI-BA11-58X*



TU-58 chassis assembly which includes rear loading **BPA82** eight slot dual width backplane/card guide assembly with removable 120 ohm termination. Triple output regulated switching power supply with +5v @ 18a, +12v @ 4a and −12v @ 1a. Two cooling fans for 200 cfm air circulation. Unit includes dual front loading TU-58 cartridge tape drives and controller with a combined storage capacity of 500K bytes. Self-contained power on/off and failure sequencing. Illuminated front panel switches to indicate when DC power is ON, and when system is in HALT mode or LTC is enabled. Illuminated RUN Indicator. Rocker style combined circuit breaker AC on/off switch is located on rear of chasssis. Strain relief for rear exiting cables. RETMA rack mountable. Dimensions: 5¼" high, 19" wide and 18" deep. Weight: 40 pounds.

MLSI-BA11-580X*

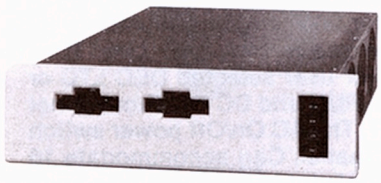
TU-58 chassis assembly which includes **BPA84** backplane to hold 16 dual size boards or 8 quad size boards. Triple output regulated switching power supply with +5v @ 18a, +12v @ 4a, and −12v @ 1 a RETMA rack mountable. Unit includes dual front loading TU-58 cartridge tape drives and controller with combined storage capacity to 500K bytes. Self-contained power on/off and failure sequencing, also front panel switches for system operation. Dimensions: 5¼" high by 19" wide by 18" deep. Weight 40 pounds.

MLSI-BA11-581X*

TU-58 chassis assembly which includes **BPA84-A** quad width backplane to hold 16 dual or 8 quad boards. Backplane can accommodate DEC RL01 or RL02 disk controller. All other features are the same as the MLSI-BA11-580X.

LSI-11 BOX SYSTEMS

MLSI-11/03-58X*



LSI-11/2-TU-58 system which includes **LSI-11/2 CPU, 2 asynchronous serial ports,** line time clock, bootstrap, **32KB of RAM memory** plus **dual TU-58** cartridge tape units with combined storage capacity of 500KB. One of the asynchronous serial ports is dedicated to the TU-58 controller. Unit is 5¼" high RETMA rack mountable system box which provides dual width 8 slot **BPA82** card cage with removable bus termination, triple output regulated switching power supply and self-contained power on/off and failure sequencing with front panel system control switches. See MLSI-BA11-58X for details of chassis assembly and power capabilities.

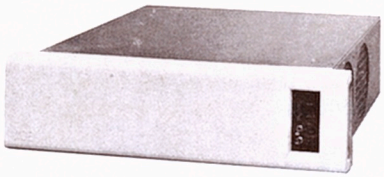
MLSI-11/03-580X*

LSI-11/2-TU-58 system similar to above, but, contains **BPA84** backplane. See MLSI-BA11-58X for details of chassis assembly and power capabilities.

MLSI-11/03-581X*

LSI-11/2-TU-58 system with **BPA84-A** backplane. All features of MLSI-BA11-58X but with a quad size backplane that will accommodate DEC's, RL01 or RL02 disk controller. See MLSI-BA11-58X for details of chassis assembly and power capabilities.

MLSI-11/03-LHBX*



LSI-11/2 system which includes **LSI-11/2 CPU, 2 asynchronous serial ports,** line time clock, bootstrap, **32KB of RAM memory** plus **EIS/FIS instruction set. Unit is 5¼" high RETMA rack mountable system box which provides 8 quad or 16 dual Q-bus slots.** Uses **BPA84** backplane. Included is triple output regulated switching power supply with power on/off and failure sequencing with front panel system control switches. See MLSI-BA11-20X for details of chassis assembly and power capabilities.

MLSI-11/03-LHX*

LSI-11/2 system same as MLSI-11/03-LHBX except slots 6 and 7 in Columns C and D are wired to accept the RL01 or RL02 disk controller. Uses **BPA84-A** backplane. See MLSI-BA11-20X for details of chassis assembly and power capabilities.

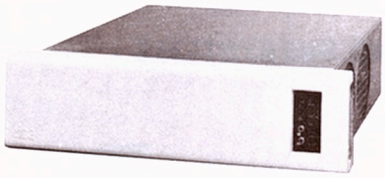
MLSI-11/03-LKBX*

LSI-11/2 system same as MLSI-11/03-LHBX except includes **64KB RAM** memory and 4 asynchronous serial ports.

MLSI-11/03-LKX*

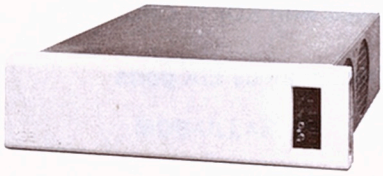
LSI-11/2 system same as MLSI-11/03-LHX except includes **64KB RAM** memory and 4 asynchronous serial ports.

MLSI-11/23-AA0X*



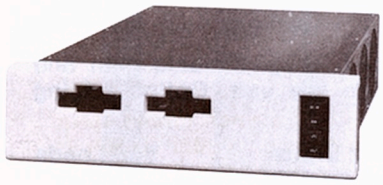
LSI-11/23 system which includes **11/23 CPU** with **memory management, 128KB of RAM memory, 4 asynchronous serial ports,** bootstrap, line time clock and bus termination. Unit is 5¼" RETMA rack mountable system box which provides 8 quad or 16 dual LSI-11 Q-bus slots **(BPA84).** Included is triple output, regulated switching power supply and power on/off and failure sequencing with front panel system control switches. See MLSI-BA11-20X for details of chassis assembly and power capabilities.

MLSI-11/23-AA1X*



LSI-11/23 system which includes **LSI-11/23 CPU** with **memory management, 128KB of RAM memory, 4 asynchronous serial ports,** bootstrap, line time clock and bus termination. Unit is 5¼" RETMA rack mountable system box which provides 8 quad or 16 dual slots with slots 6, C-D and 7, C-D wired to accept the RL01 or RL02 disk controller **(BPA84-A).** 14 slots are wired for Q-bus. Included is triple output, regulated switching power supply and power on/off and failure sequencing with front panel system control switches. See MLSI-BA11-20X for details of chassis assembly and power capabilities.

MLSI-11/23-581X*



LSI-11/23-TU-58 system which includes **LSI-11/23 CPU** with **memory management, 128KB of RAM memory, 4 asynchronous serial ports,** bootstrap, line time clock and bus terminator, plus **dual TU-58** cartridge tape units with combined storage capacity of 500KB. One of the 4 serial ports is dedicated to the TU-58 controller. Unit is 5¼" high RETMA rack mountable system box which provides 8 quad or 16 dual slots with slots 6, C-D and 7, C-D wired to accept RL01 or RL02 disk controller. 14 slots are wired for Q-bus. Uses **BPA84-A** backplane with **rear loading** capability. Included is triple output regulated switching power supply and self contained power on/off and failure sequencing with front panel system control switches. See MLSI-BA11-58X for details of chassis assembly and power capabilities.

*In the BA11 Series or the LSI-11 System Boxes, procedure is to indicate either X = 0 for 110 VAC, 60Hz operation or X = 2 for 220 VAC 50Hz, operation.

MASS STORAGE SUBSYSTEMS

MLSI-RXV21-BA	1MB double density, dual drive floppy disk subsystem with DMA controller
MLSI-DSD-440	1MB double density, dual drive floppy disk subsystem with DMA controller which occupies 1 dual slot
MLSI-DSD-480	2MB double density, double sided dual drive floppy disk subsystem with DMA controller which occupies 1 dual slot.
MLSI-DSD-880	Combination 5.2MB winchester disk drive with a 1MB floppy disk in a single 5¼" high RETMA rack mountable chassis. Complete with DMA dual size controller.
MLSI-TU58-BB	500KB dual tape cartridge subsystem. Includes two drives and interface

CABLE SUBASSEMBLIES

MDB-BC05M-2C	I/O Cable for 20ma current loop (2 ft.)
MDB-BC05C-25	I/O Cable for EIA (25 ft.) (Asynchronous/Synchronous)
MDB-BC20N-05	Null modem cable for DLV11-J or MXV11-AC
GPIO-50-D-4	Double-ended jumber cable assembly. 50 conductor flat ribbon cable lengths 4 feet.

The following cable have a single edge connector with 10 feet of unterminated cable:

MDB-BC050Z-10	50 conductor ribbon cable
MDB-BC040Z-10	40 conductor ribbon cable
MDB-BC034Z-10	34 conductor ribbon cable
MDB-BC026Z-10	26 conductor ribbon cable
MDB-BC020Z-10	20 conductor ribbon cable

Longer cable length is obtainable.

LSI-11 SYSTEM MICROPROCESSOR MODULES AND SUBSYSTEMS

MLSI-KDF11-HF	LSI-11/23 CPU with MMU plus 128KB RAM
MLSI-KDF11-HK	LSI-11/23 CPU with MMU plus 256KB RAM
MLSI-KD11-HA	LSI-11/2 CPU (memory not included)
MLSI-KD11-HD	LSI-11/2 CPU with 64KB RAM
MLSI-KEF11-AA	Floating point unit for LSI-11/23 (when purchased with KDF11 CPU)
MLSI-KEV11	Extended arithmetic unit for LSI-11/2
MLSI-MXV11-AC	Multi-Function Module with 32KB RAM & 2 asynchronous EIA ports
MLSI-MXV11-A2	Bootstrap PROM for MLSI-MXV11-AC
MLSI-MSV-07	32KB RAM expansion module
MLSI-MSV-10	64KB RAM expansion module

ANALOG/DIGITAL SUBSYSTEMS

MLSI-DT2766
(AAV11-A type)

Four-channel, 12-bit DAC subsystem. Optional programmable gain and 20ma inputs. Options must be ordered with the unit. Option pricing available at time of order.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	1.5a	1 max

MLSI-DT2762
(ADV11-A type)

Sixteen-channel, 12-bit ADC subsystem. Optional programmable gain and 100KHz throughput. Options must be ordered with the unit. Option pricing available at time of order.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual slot	1a	1 max

MLSI-DT2781

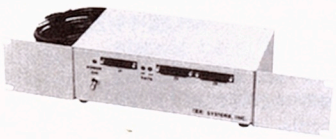
Analog I/O subsystems, contains 16-channel, 12-bit A/D plus two 12-bit D/A channels. Optional programmable gain available at time of order. Option pricing available at time of order.

Mounting Code	Power Drawn Amps @ +5V	Bus Loads Drawn
Dual Slot	1.5 a	1 max

MLSI-DT701-20-TMA Universal screw terminal distribution panel. Provides for connection of up to 50 separate inputs. Includes thermocouple cold reference junction compensation circuit plus area for user to mount own signal conditioning or protection circuits. Enclosed RETMA rack mountable panel 19" wide and 5" high.

UNIVERSAL LONG LINE ADAPTER BOXES

MDB-ULLAB-01



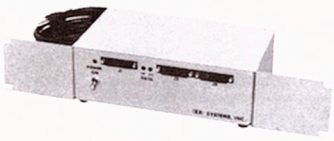
Universal EIA RS-232-C serial long line adapter box that contains **six** RS422 (differential long line) receivers that are converted to RS232-C drivers and **six** RS232-C receivers that are converted to RS422 drivers. Used to adapt any RS232 device to an RS422 interface that can be driven up to 4,000 feet. Also used to convert RS422 signals to RS232-C levels. Can be used with MLSI-DLV11-F, which provides RS422 capabilities.

Power Requirements

115VAC at .5a

MDB-ULLAB-02 **Same as MDB-ULLAB-01 except** contains **12** driver circuits and **12** receiver circuits.

MDB-ULLAB-03

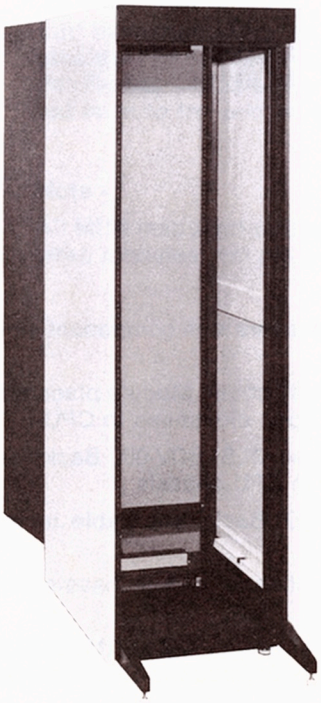


Universal TTL parallel long line adapter box that contains **twelve** RS422 (differential, long line) receivers that are converted to TTL drivers and **twelve** TTL receivers that are converted to RS422 drivers. Used to convert most devices with a TLL interface to a differential interface for long line drive capability. Used to provide long line capability for MDB line printer controllers for the LSI-11 computer. Optional input adapter cables also required.

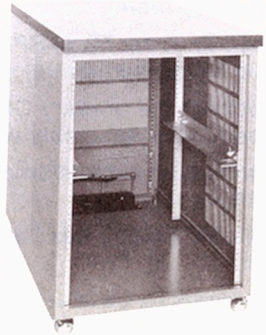
Power Requirements

115VAC at .5a

MDB-ULLAB-RM **Rack mount option** for above units to allow mounting to standard 19 inch RETMA rack. (Add-RM to above part numbers to specify this option.) Photos show this option installed.



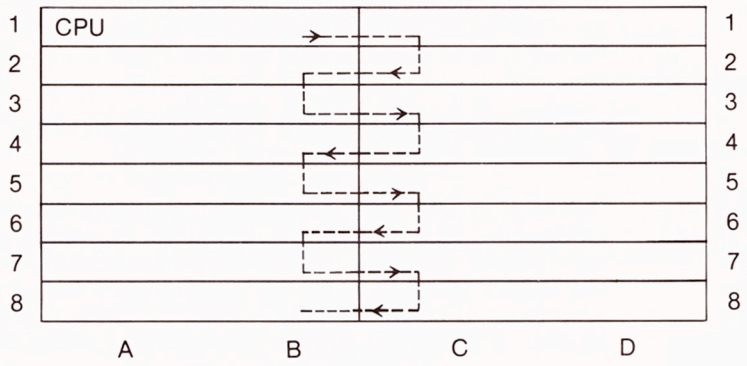
MLSI-H961-A



MLSI-H984

BACKPLANE CONFIGURATIONS

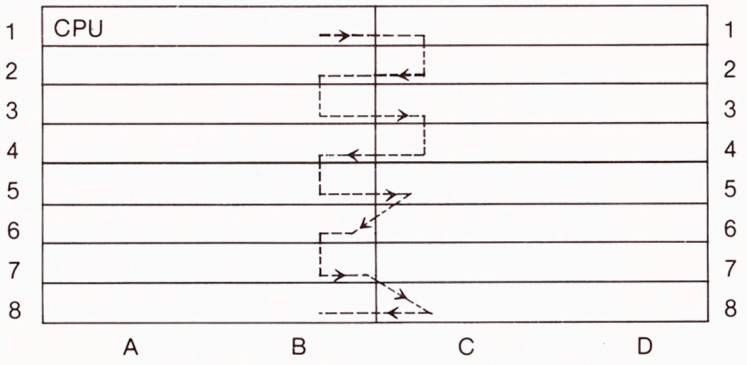
MLSI-BPA84



MLSI-BPA84 USED IN

MLSI-BA11-20X
MLSI-BA11-30X
MSLI-11/03-LHBX
MLSI-11/O3-LKBX
MLSI-BA11-580X
MLSI-11/23-AA0X
MLSI-11/23-580X

MLSI-BPA84-A

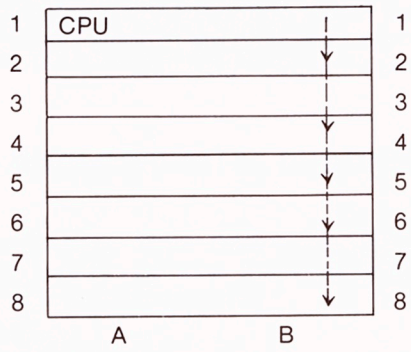


MLSI-BPA84-A USED IN

MLSI-BA11-21X
MLSI-BA11-31X
MLSI-11/03-LHX
MLSI-11/03-LKX
MLSI-11/23-AA1X
MLSI-11/23-581X

Slots 6C-D and 7C-D used for RL01 or RL02 Disk Controller.

MLSI-BPA82



MLSI-BPA82 USED IN

MLSI-BA11-58X
MLSI-11/03-58X

APPLICATION NOTES

1. Chassis are shown as viewed from front, or module plug-in side of chassis with component sides of modules facing up.
2. Dashed lines for MLSI-BPA84 and BPA84-A show Q bus priority daisy chain. Module placement should follow sequential slots. Priority for modules is dependent upon closeness to CPU.
3. To connect the primary CPU backplane to an expansion chassis, use MLSI-BCV-000 Backplane Jumper/Cable Assembly with resistor termination board installed in CPU chassis.
4. To connect an expansion chassis to a third chassis, use MLSI-BCV-001 Backplane/Cable Jumper Assembly.
5. Use MLSI-TEV-003 Bus Terminator Module to terminate expansion chassis, or 3rd chassis. (End of Bus)
6. If necessary to leave an open slot in priority daisy chain a Bus Grant Continuity Card (MLSI-BGC) must be placed in either A or C slot only of vacant location. This is to pass the DMA and Bus Grant onto the next board.

Note 1:

Certain MDB LSI-11 interfaces have been designed to run with PDP-11 Unibus software drivers for the equivalent DEC Unibus product. This unique feature is of great advantage when transposing software (particularly under RSX-11M) from a Unibus system to an LSI-11/2 or LSI-11/23 based system. Consult MDB factory for specific programming characteristics of individual LSI-11 interface products.

Note 2:

Buffer Ready/Printer Busy monitor circuit is user strappable to allow transmission of data to printers or other devices which contain an RS-232 serial interface, **and** which signify a buffer full. When the buffer status line on the peripheral goes from SPACE (+12V) to MARK (−12V), or vice versa if so selected, the module will stop sending data until the peripheral signifies that data transmission should be started again. Since the MDB module is double buffered, when used with printers or other peripherals that signify Buffer completely full, a character will be lost unless a single pad character (NUL) is transmitted after every CR/LF command. This requirement is not necessary for devices that signify buffer almost full, or that can accept a character after reporting their buffer full status.

Note 3:

PrinTest: When actuated by an edge mounted toggle switch on the controller, or by a remote signal, the controller will transmit to an operational printer a pre-formatted test pattern, shown below, which exercises the printer interface portion of the controller, the interconnection cable, and all parameters of the printer except VFU functions. The PrinTest feature can be also activated remotely if a low-going TTL signal is applied to pin 17 at the controller's I/O connector. The wire is in the cable and is already connected to pin 36 on Centronics cables at the printer interface connector. It is not connected to any pin on other types of interface connectors.

```
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
PRINTEST !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~ 123456789      PRINTEST
```

Loop Back: When actuated by an edge mounted toggle switch on the controller, the controller will react to the CPU as a complete printer sub-system, even though a printer may not be connected to the module. This feature in conjunction with PrinTest and the LED indicators can be used for complete controller verification of proper operation or for trouble shooting of the module without the normal association of printer noise and wasted paper.

LED Indicators: Edge mounted LEDs give visual indication of data that is present on the 8 data line plus data strobe functions. An Error LED presents the logical ANDing of On-Line (Select) and Ready (No Faults) status signals originating from the printer. This Error/Ready signal can also be transmitted back to the printer. (Pin 33 on Centronics connector.)

Note 4:

All MDB manufactured modules are of UL approved FR4 material. All holes, including those on wire-wrap modules, are plated through.

MDB SYSTEMS COMPUTER INTERFACE PRODUCTS

	DEC LSI-11/2* LSI-11/23*	DEC PDP-11* VAX-11/780*	DEC PDP-8* A,E,F,M	DATA GENERAL** NOVA ECLIPSE	PERKIN-ELMER (Interdata)	H-P 2100 21MX 1000	IBM SERIES/1	INTEL MULTIBUS
PERIPHERAL DEVICE CONTROLLERS Line Printers	●	●	●	●	●	●	●	
Line Printers Long Line (RS-422)		●		●	●	●		
Card Readers	●	●		●	●			
Incremental X-Y Plotters	●	●						
Electrostatic Printer/Plotters	●	●						
Paper Tape Reader/Punches	●	●						
IEEE Interfaces	●	●			●			
COMMUNICATIONS MODULES Asynchronous Interfaces (RS-232, 20ma RS-422)	●	●		●	●		●	
Synchronous Interfaces (RS-232)	●	●						
Asynchronous Multiplexors (RS-232, 20ma)	●	●		●				
DMA Parallel Terminal Interfaces (RS-422)		●						
INTERPROCESSOR LINKS Differential I/O Circuitry (DMA)	●	●						
Optical Isolator Circuitry (DMA)	●	●						
TTL Circuitry (Programmed I/O)		●						
BUS FOUNDATION MODULES WITH WIRE WRAP DMA BUS Structure	●	●		●				
Programmed I/O Structure	●	●		●	●			●
SYSTEMS MODULES DMA BUS Structure	●	●		●				
Programmed I/O Structure	●	●		●	●			
Clock Modules	●	●		●	●			
PROM BOARDS Bootstrap Loader	●	●						
Standard PROM Boards	●	●						
PROM Programmer Boards	●	●						
WIRE WRAP MODULES Pins Only	●	●	●	●	●		●	●
Pins and Sockets	●	●	●	●	●			
CHASSIS/BACKPLANES/ENCLOSURES Computer Chassis-Backplane	●							
Expansion Chassis	●			●				
BUS Terminators	●	●		●				
CABINETS Roll-Around	●							
Stand Alone	●							
COMPUTER BOX SYSTEMS	●							
COMPUTERS	●							
MEMORIES	●							
CABLES General Purpose	●	●	●	●	●		●	●
Communications	●	●		●	●		●	
ACCESSORIES Board Extenders				●			●	●
Mounting Kits					●			
Power Supplies	●			●				
A-D/D-A CONVERTERS	●	●						
COMMUNICATIONS PANELS	●	●		●				
BUS Grant Cards	●	●						
UNIVERSAL LONG LINE ADAPTERS	●	●	●	●	●	●	●	●

* TM Digital Equipment Corporation ** Data General is a computer manufacturer not related to MDB

MDB U.S. SALES OFFICES AND FOREIGN DISTRIBUTORS

UNITED STATES:
Central Northeast: 216/247-2630 ☐ Northern California: 408/733-3950 ☐ Mid-Atlantic: 703/893-0753 ☐ New England: 617/444-1800 ☐ NY/NJ/Eastern Pennsylvania: 201/265-0043 ☐ Mid-Western: 312/266-1032 ☐ Southeastern: 305/254-2511 ☐ Southwest Central: 713/376-3973 ☐ Southwestern: 214/559-3890 ☐ Hawaii: 808/524-8633 ☐ Southern California: 714/998-6900 ☐
FOREIGN COUNTRIES:
MDB has distributors in the following countries:
Argentina ☐ Australia ☐ Belgium ☐ Canada ☐ Denmark ☐ Finland ☐ France ☐ Greece ☐ Hong Kong ☐ India ☐ Israel ☐ Italy ☐ Japan ☐ Luxembourg ☐ Malaysia ☐ Netherlands ☐ New Zealand ☐ Norway ☐ Philippines ☐ Singapore ☐ South Africa ☐ Sweden ☐ Switzerland & Liechtenstein ☐ United Kingdom ☐ W. Germany ☐ Taiwan ☐

MDB INTERFACE PRODUCTS for use with Digital LSI*-11/2 & 11/23 Computers PRICE LIST

MDB LSI-11 PRICE LIST

MDB PART NUMBER	UNIT QUANTITIES 1-9	10-24	25-49	50-99	100-up
LSI-11 COMMUNICATIONS/TERMINAL INTERFACE MODULES					
(Asynchronous Serial Interfaces)					
MLSI-DLV11	475	404	356	318	309
MLSI-DLV11-E	370	315	278	248	241
MLSI-DLV11-F	350	297	263	235	228
MLSI-DLV11-FX	350	297	263	235	228
MLSI-DLV11-J**	525	525	525	525	525
Optional cables available.					
(Single Line Synchronous Interfaces)					
MLSI-DUV11	700	595	525	469	455
MLSI-DUPV11	950	855	808	779	760
Above modules include cables.					
(8 and 16 Port Asynchronous Multiplexors)					
MLSI-DZ11-A	1750	1575	1490	1435	1400
MLSI-DZ11-AC	1900	1710	1615	1558	1520
MLSI-DZ11-B	1350	1215	1150	1110	1080
MLSI-DZ11-E	2800	2520	2380	2290	2240
LSI-11 INTERPROCESSOR LINK SUBSYSTEMS					
MLSI-DA11BOI	3295	2966	2801	2702	2636
MDB-MLSI-DA11BOI	4050	3645	3443	3321	3240
Above modules include 50' cable pairs.					
LSI-11 PERIPHERAL DEVICE INTERFACE MODULES					
MLSI-LP11 (Centronics/Dataproducts)	475	452	428	404	380
MLSI-LP11-A (Data Printer)	650	585	553	533	520
MLSI-LV11 (Versatec)	975	878	829	800	780
MLSI-XYV11 (Calcomp 500 Series)	700	630	595	574	560
MLSI-CR11 (Documation/Truedata)	850	765	723	697	680
MLSI-PC11 (Remex, EECO, etc.)	750	675	638	615	600
Above modules include 15' cables.					
LSI-11 SYSTEM MODULES					
MLSI-IBV11	725	616	544	485	471
MLSI-DRV11-C	275	234	206	184	179
MLSI-KW11-P	725	616	544	485	471
MLSI-DR11-B	650	553	488	436	423
MLSI-DRV11-B	670	570	503	442	429
MLSI-BCV-000	310	264	233	208	202
MLSI-BCV-001	230	196	173	154	130
MLSI-TEV-003	110	94	83	74	72
MLSI-REV11-A**	320	320	320	320	320
MLSI-REV11-C**	320	320	320	320	320
MLSI-BGC**	15	15	15	15	15

Effective Date: 03/1/81
Prices: F.O.B. Orange, California,
subject to change without notice.

MDB SYSTEMS INC.
1995 N. Batavia Street
Orange, California 92665
714-998-6900
TWX: 910-593-1339

Printed in U.S.A.
L/PL/5M/3-81 REPRINT 5-81

* TM Digital Equipment Corp.

MDB LSI-11 PRICE LIST (Continued)

MDB PART NUMBER		UNIT QUANTITIES 1-9	10-24	25-49	50-99	100-up
GENERAL PURPOSE LSI-11 (P)ROM MEMORY MODULES						
MLSI-MRV-000		195	166	146	131	127
MLSI-MRV-001		195	166	146	131	127
MLSI-MRV-002		195	166	146	131	127
MLSI-MRV-003		195	166	146	131	127
MLSI-MRV-004		550	495	468	451	440
MLSI-MRV-005		350	315	298	287	280
Above modules do not include (P)ROMS.						
LSI-11 GENERAL PURPOSE BUS INTERFACE FOUNDATION AND WIREWRAP MODULES						
MLSI-1710		225	191	169	151	146
MLSI-DRV11-P		325	276	244	218	211
MLSI-11B		650	585	553	533	520
MLSI-WWB1		75	64	57	51	49
MDB-W9501		175	149	132	118	114
Optional cable subassemblies available.						
LSI-11 SYSTEM CHASSIS AND ENCLOSURE SUBASSEMBLIES						
MLSI-BPA82*		275	234	207	185	179
MLSI-BPA84*		425	362	319	285	277
MLSI-BPA84-A*		565	481	424	379	368
MLSI-H984**		675	675	675	675	675
MLSI-H961-A**		1490	1490	1490	1490	1490
MLSI-BA11-20X	(BPA84)	1990	1692	1493	1334	1294
MLSI-BA11-21X	(BPA84-A)	2130	1811	1598	1428	1385
MLSI-BA11-30X	(BPA84)	1550	1318	1163	1039	1008
MLSI-BA11-31X	(BPA84-A)	1690	1437	1268	1133	1099
MLSI-BA11-58X	(BPA82)	3050	2593	2288	2044	1983
MLSI-BA11-580X	(BPA84)	3330	2805	2475	2211	2145
MLSI-BA11-581X	(BPA84-A)	3440	2924	2580	2305	2236
	Backplane type used in chassis.					
LSI-11 BOX SYSTEMS						
MLSI-11/03-58X	(BPA82)	4650	4185	3953	3813	3720
MLSI-11/03-580X	(BPA84)	4800	4320	4080	3936	3840
MLSI-11/03-581X	(BPA84-A)	5040	4536	4284	4133	4032
MLSI-11/03-LHBX	(BPA84)	3440	3225	3010	2795	2795
MLSI-11/03-LHX	(BPA84-A)	3580	3356	3133	2909	2909
MLSI-11/03-LKBX	(BPA84)	3680	3450	3220	2990	2990
MLSI-11/03-LKX	(BPA84-A)	3820	3581	3343	3104	3104
MLSI-11/23-AA0X	(BPA84)	6080	5700	5320	4940	4940
MLSI-11/23-AA1X	(BPA84-A)	6220	5831	5443	5054	5054
MLSI-11/23-581X	(BPA84-A)	7530	6777	6401	6175	6024
MASS STORAGE SUBSYSTEMS						
MLSI-RXV21-BA**		3695	3695	3695	3695	3695
MLSI-DSD-440**		3795	3795	3795	3795	3795
MLSI-DSD-480**		4795	4795	4795	4795	4795
MLSI-DSD-880**		7495	7495	7495	7495	7495
MLSI-TU-58-BB**		795	795	795	795	795

***BPA82** 8 dual size boards, **BPA84** 16 dual or 8 quad, **BPA84-A** slots 6 C-D & 7 C-D wired for hard disk controller, all others for Q Bus.

Where used, X designates power requirements. 0=115V 60 Hz, 1=230 V 50 Hz.

MDB LSI-11 PRICE LIST (Continued)

MDB PART NUMBER	UNIT QUANTITIES 1-9	10-24	25-49	50-99	100-up
CABLE SUBASSEMBLIES					
MDB-BCO5M-2C** (Current Loop)	50	50	50	50	50
MDB-BCO5C-25** (RS-232)	120	120	120	120	120
MDB-BC2ON-05** (Null Modem)	40	40	40	40	40
GPIO-50-D4**	45	45	45	45	45
MDB-BCO50Z-10**	85	85	85	85	85
MDB-BCO40Z-10**	75	75	75	75	75
MDB-BCO34Z-10**	60	60	60	60	60
MDB-BCO26Z-10**	50	50	50	50	50
MDB-BCO20Z-10**	40	40	40	40	40
LSI-11 MICROPROCESSOR MODULES AND SUBSYSTEMS					
MLSI-KDF11-HF**	4050	3690	3690	3690	3690
MLSI-KDF11-HK**	5445	4961	4961	4961	4961
MLSI-KD11-HA**	695	695	695	695	695
MLSI-KD11-HD**	1521	1386	1386	1386	1386
MLSI-KEF11-AA**	378	345	345	345	345
MLSI-KEV11**	180	164	164	164	164
MLSI-MXV11-AC**	900	820	820	820	820
MLSI-MXV11-A2**	55	55	55	55	55
MLSI-MSV-07**	945	861	861	861	861
MLSI-MSV-10**	1080	984	984	984	984
ANALOG/DIGITAL SUBSYSTEMS					
MLSI-DT2766**	995	995	995	995	995
MLSI-DT2762**	995	995	995	995	995
MLSI-DT2781**	1295	1295	1295	1295	1295
MLSI-DT701-20-TMA**	475	475	475	475	475
UNIVERSAL LONG LINE ADAPTER BOXES					
MDB-ULLAB-01	525	475	447	431	420
MDB-ULLAB-02	575	518	489	478	460
MDB-ULLAB-03	525	475	447	431	420
MDB-ULLAB-RM (1)**	25	23	22	21	20

(1) Option to ULLAB-01,-02,-03

General Purpose IO and other cables can be obtained in longer cable lengths at the additional price per foot indicated below.

MDB-BCO50Z	$3.00	MDB-BCO26Z	$1.50	Line Printer Cables	$3.00
MDB-BCO40Z	$2.50	MDB-BCO20Z	$1.25	Card Reader Cables	$3.00
MDB-BCO34Z	$2.00			Inter-Processor Links	$5.00 per cable pair.

Total number of modules purchased at one time determines Unit Quantity price for the specific purchase order. For example, ten identical or ten different discountable products will be invoiced at their respective quantity 10-24 price.

An **Accumulative Quantity Discount Agreement** is available to cover a one year period. This provides price protection for one year and ensures the lowest price as each quantity level is attained. Contact factory or your local MDB Sales Office for details.

**These items do not accumulate toward the quantity discount level.

MDB SYSTEMS COMPUTER INTERFACE PRODUCTS

	DEC LSI-11/2* LSI-11/23*	DEC PDP-11* VAX-11/780*	DEC PDP-8* A,E,F,M	DATA GENERAL** NOVA ECLIPSE	PERKIN-ELMER (Interdata)	H-P 2100 21MX 1000	IBM SERIES/1	INTEL MULTIBUS
PERIPHERAL DEVICE CONTROLLERS Line Printers	●	●	●	●	●	●	●	
Line Printers Long Line (RS-422)		●		●	●	●		
Card Readers	●	●		●	●			
Incremental X-Y Plotters	●	●						
Electrostatic Printer/Plotters	●	●						
Paper Tape Reader/Punches	●	●						
IEEE Interfaces	●	●			●			
COMMUNICATIONS MODULES Asynchronous Interfaces (RS-232, 20ma RS-422)	●	●		●	●		●	
Synchronous Interfaces (RS-232)	●	●						
Asynchronous Multiplexors (RS-232, 20ma)	●	●		●				
DMA Parallel Terminal Interfaces (RS-422)		●						
INTERPROCESSOR LINKS Differential I/O Circuitry (DMA)	●	●						
Optical Isolator Circuitry (DMA)	●	●						
TTL Circuitry (Programmed I/O)		●						
BUS FOUNDATION MODULES WITH WIRE WRAP DMA BUS Structure	●	●		●				
Programmed I/O Structure	●	●		●	●			●
SYSTEMS MODULES DMA BUS Structure	●	●		●				
Programmed I/O Structure	●	●		●	●			
Clock Modules	●	●		●	●			
PROM BOARDS Bootstrap Loader	●	●						
Standard PROM Boards	●	●						
PROM Programmer Boards	●	●						
WIRE WRAP MODULES Pins Only	●	●	●	●	●		●	●
Pins and Sockets	●	●	●	●	●			
CHASSIS/BACKPLANES/ENCLOSURES Computer Chassis-Backplane	●							
Expansion Chassis	●			●				
BUS Terminators	●	●		●				
CABINETS Roll-Around	●							
Stand Alone	●							
COMPUTER BOX SYSTEMS	●							
COMPUTERS	●							
MEMORIES	●							
CABLES General Purpose	●	●	●	●	●		●	●
Communications	●	●		●	●		●	
ACCESSORIES Board Extenders				●			●	●
Mounting Kits					●			
Power Supplies	●			●				
A-D/D-A CONVERTERS	●	●						
COMMUNICATIONS PANELS	●	●		●				
BUS Grant Cards	●	●						
UNIVERSAL LONG LINE ADAPTERS	●	●	●	●	●	●	●	●

* TM Digital Equipment Corporation ** Data General is a computer manufacturer not related to MDB

MDB U.S. SALES OFFICES AND FOREIGN DISTRIBUTORS

UNITED STATES:
Central Northeast: 216/247-2630 ☐ Northern California: 408/733-3950 ☐ Mid-Atlantic: 703/893-0753 ☐ New England: 617/444-1800 ☐ NY/NJ/Eastern Pennsylvania: 201/265-0043 ☐ Mid-Western: 312/266-1032 ☐ Southeastern: 305/254-2511 ☐ Southwest Central: 713/376-3973 ☐ Southwestern: 214/559-3890 ☐ Hawaii: 808/524-8633 ☐ Southern California: 714/998-6900 ☐
FOREIGN COUNTRIES:
MDB has distributors in the following countries:
Argentina ☐ Australia ☐ Belgium ☐ Canada ☐ Denmark ☐ Finland ☐ France ☐ Greece ☐ Hong Kong ☐ India ☐ Israel ☐ Italy ☐ Japan ☐ Luxembourg ☐ Malaysia ☐ Netherlands ☐ New Zealand ☐ Norway ☐ Philippines ☐ Singapore ☐ South Africa ☐ Sweden ☐ Switzerland & Liechtenstein ☐ United Kingdom ☐ W. Germany ☐ Taiwan ☐

WHY PASCAL
****

Many computer manufacturers, consulting firms, and large companies say that Pascal is the "language of the future" here today. Pascal has been dubbed the "BASIC" of the future and "PL/1" done right. Why, because Pascal allows the programmer to define his problem in a clear, concise way. It is an easy language to learn, extremely readable, easy to modify, yet fully supports advanced programming techniques such as data structuring, recursion, extensibility, and user defined data types. Pascal is an ideal, block-structured programming language that inherently forces structured programming techniques, top-down design, and good programming practices.

Using Pascal with all of its advanced features equates to a dramatic increase in programmer productivity with an even more dramatic decrease in the time spent in program maintenance and modification.

> "This directly equates to a dramatic decrease in program development and maintenance costs in some cases by a factor of 10 over the total lifecycle of the software system."

Computer scientist, Dr. Peter Naur, estimated that a new programming language would need to provide a factor-of-ten cost advantage to supplant its predecessors. Is Pascal a good enough technology to cut overall software costs 80 to 90 percent over the life cycle of a software application. There is a good chance that the answer is yes. Univac, when touting its Pascal compiler for its minicomputers claims that Pascal programming is ten times as cost effective as COBOL programming.

Recently a large computer manufacturer did an extensive survey and rated Pascal against the three most popular programming languages FORTRAN, COBOL, and BASIC. A rating of ten does not imply perfection, only that the language is much better than the four others in the category. Here are the results:

	PASCAL	**FORTRAN**	**BASIC**	**COBOL**
• Applicability for complex software	10	6	3	7
• Applicability for small programs	8	7	10	3
• Ease of learning the language	7	8	10	3
• Supports top-down design	10	5	3	4
• Structured programming control constructs	10	4	4	6
• Flexibility of data types	10	5	1	7
• Compiler Error Checking & Consistency	10	5	4	7
• Language has effective problem solving	10	6	1	7
• Readability of large programs	10	4	2	8
• Language supports transportability	10	7	4	8

Pascal distributed as a portable "P" Code compiler is currently available on over seventy different computer systems.

- Why is Pascal so wide spread?
  - Why are so many companies excited about it?
    - Why does Pascal dramatically reduce program development and software maintenance costs?

Here are a few of the reasons why:

- Pascal was designed from the start as a structured programming language. Structured programming is a proven method for significantly reducing program development costs. Pascal offers the control and data structures for implementing computer algorithms and applications unmatched by any other language.

- Pascal's main innovation is the introduction of much more sophisticiated data structuring facilities. It provides as standard arrays, records, sets, files, and pointers. Pascal even allows the programmer to define his own data types such as complex numbers and multi-dimensioned arrays to allow him to create the data structures he needs to accurately model the problem that he is trying to solve.

- Pascal provides unmatched error checking and type checking during compilation to catch subtle errors that would normally go undetected with other language compilers.

- Pascal programs use English words and clear structures that are much easier to understand, change, maintain, and correct. Software, unlike hardware is modified continually to accommodate user needs. Therefore, it is essential to have very readable, easy to modify programs with a well defined interface between the software modules.

- Pascal portability across many different computer systems allows applications programs, system software, and utilities developed in Pascal to be distributed over many different systems. The portability allows running the same software on many different computer architectures drastically reducing implementation costs on new computer systems.

- Pascal has "extensibility" which means that the programmer does not have to pay a price for its powerful, yet simplistic, concise structure. With Pascal the programmer can create exactly what he needs. Pascal does not try to provide "everything for everyone" such as a complex language as "PL/1." Pascal is small, fast, and complete with all the capabilities that a programmer needs.

Pascal simplicity, as well as its power for expressing complex algorithms, establishes it as a language well suited for both writing complex applications and system software as well as an excellent teaching tool for teaching correct programming techniques. Pascal has been or is replacing assembly language for real time and process control applications. Pascal has become the "lingua franca" in computer science departments. Nearly every major University computer science curriculum uses Pascal as its primary teaching language. They are training tommorrow's systems analysts, scientists, and businessmen. That is why they are teaching Pascal, today.

FEATURES & ENHANCEMENTS TO THE ACD UCSD PASCAL VERSION III.0 FOR THE PDQ-3

The III.I release contains several useful features not provided in other releases of the UCSD PASCAL system. The most significant feature is the new concurrency system. The III.I system is full interrupt driven, allowing multiple-terminal applications to be written (assuming the presence of supporting hardware). Console type-ahead is implemented, and there is a system clock, accessed through the UCSD PASCAL Timer intrinsic. The PDQ-3 Programmer's Manual will explain the concurrency system, but there is a short primer at the end of this document in lieu of that manual.

Also included in this release are certain bug fixes and speed improvements; notably in MOVELEFT and MOVERIGHT. There is a new Submit file processor (X.CODE) and new utilities (FORMAT.CODE, BAD.BLOCKS.CODE, etc). The PRINTER.CODE utility causes files to be queued for printing by the system spooler. This spooler can spool to any serial unit, including REMOUT: (#8:), PRINTER: (#6:), and REMOUT1 through REMOUT4 (#14: through #20:).

Advanced Computer Design is offering the Advanced System Editor (ASE) as an alternative to the Western Digital SYSTEM.EDITOR provided with the ACD PASCAL System. The ASE provides several keystroke optimizations, special key macro facitilities, recursive editing, extended exchange modes (with features similar to the Emax Editor developed at M.I.T.), the ability to edit LARGE files, and many bug fixes. The ASE comes with 50 pages of documentation, costs $200.00, and saves between 25% and 50% of your editing time.

By July 1, Advanced Computer Design will offer the ACD OEM Unit package. It will contain UCSD PASCAL Units that provide useful operating system interfacing functions such as wildcard directory lookups (where the program passes a filename with imbedded wildcard(s) and the unit returns a linked list of records containing information on filenames that match the original filename), integer/real to string and string to integer/real conversions, screen and keyboard control units, and concurrency support functions (task status, time delays, etc). The specifications of these units will be available at the USUS meeting in Philadelphia on June 27.

Also on July 1, Advanced Computer Design will offer subscriptions to the USUS library and to the Pascal User's Group (PUG) library. These subscriptions include membership in these organizations and copies of the available library programs on diskettes. ACD will provide this service at cost to all PDQ-3 customers.

By June 15, Advanced Computer Design will be able to create accounts on the TELENET network. The TELENET is available on a dial-in basis in most areas and is used to send and receive electronic mail. ACD plans to use the TELENET to support the PDQ-3 system and to promote communication among UCSD PASCAL users in general. The UCSD User's Group maintains accounts on TELENET as do many major UCSD PASCAL vendors. To use the TELENET, you need a modem and a modem card. An account and equipment

($995 for the modem and $339 for the card and cable) can be obrained from ACD. We strongly recommend the use of this resource.

Also as of June, Advanced Computer Design is implementing a new software support policy. Consultations with the ACD software group will typically occur either by phone or on the TELENET. TELENET communications will be answered periodically (once or twice a day), TELENET users requiring immediate service may phone ACD and receive immediate service.

Because Pascal is such an excellent language for developing complex applications software and sophisticated software tools as well as other language compilers and operating systems, there is currently available or under development:

- A wide variety of sophisticated data base management systems, with features of such popular systems as Adabase, Total, and Reality some of which will support a multi-user applications capability.
- Integrated accounting and financial management systems developed around these powerful data base management systems.
- Complete medical billing and office management systems that cover a complete spectrum of medical applications.
- Legal billing, estate planning, and word processing system.
- Sophisticated financial and econometric modeling systems.
- CPA client accounting systems.
- Complete travel agency accounting and office management systems.
- Easy to use word processing systems.

All of our customers that have been developing sophisticated computer applications with the PDQ-3 have found that the PDQ-3 with its high speed program execution and two to three times the memory space of comparable systems, and with its multi-tasking capabilities have been able to develop applications that previously could only run on large systems. The PDQ-3 with the tremendous expansion capabilities of the LSI-11 "Q" bus allowsthe PDQ-3 to grow with their applications.

C O N C U R R E N C Y

Concurrency is defined as the simultaneous execution of a number of activities. Most computer systems simulate concurrency by implementing "virtual machines" on the physical machine; a virtual machine is assigned its own processor and memory. Each of the concurrent activities executes on its own virtual machine, leaving the physical machine responsible for simulating concurrent execution of the virtual machines. Virtual machines are generally referred to as Concurrency in UCSD Pascal is restricted to concurrent execution of routines declared in a user program; thus, while a program may initiate any number of tasks, the tasks must finish executing before the program is allowed to terminate.

The PDQ-3 is a single processor system; therefore, concurrency is simulated by sharing the processor among tasks. Processor sharing is accomplished by allocating each task a period of time during which it can execute, and then switching the processor to another task; this action is known as a task switch. The task executing on the processor is called the current task, while tasks waiting for processor time are called ready to run tasks. Execution states (i.e. processor register values) describing ready to run tasks are stored in a system structure known as the ready queue.

Semaphores are special variables used for task synchronization. Semaphores are used both to prevent tasks from executing until an event occurs, and to signal occurrences of events. Tasks stopped by a semaphore are called suspended tasks. Execution states describing suspended tasks are stored in a semaphore's wait queue.

Concurrency is useful in maximizing processor utilization during I/O operations. In systems lacking concurrency, the processor waits for I/O operations to finish before proceeding; processor time is consumed by a wait loop during the I/O operation. On the PDQ-3, tasks performing I/O operations are suspended until the operation is finished, allowing other tasks to resume execution; thus, processor time need not be wasted during I/O operations.

Tasks possess the following attributes: process, task identifier, task stack size, and task priority.

The primary attribute of a task is the code it executes; tasks may only execute routines declared as processes. Processes are described below. Each task is assigned a unique identifying value when it is created; this value may be retained in a task identifier, thus allowing tasks to be distinguished from each other. Task identifiers are described below. The amount of memory allocated for a new task is determined by the task's stack size. Task stacks are described below. The last (but not least) attribute is task priority; a task's priority value determines its ability to obtain processor time in the presence of other tasks. Task priorities are described below.

Task attributes are assigned to a task when it is initiated. A task is initiated with the UCSD intrinsic START, which has the following form:

```
START ( <PROCESS call> [, <processid variable>
                       [, <stacksize expression>
```

[, <priority expression>]]]);

The main parameter to START is a PROCESS call; it resembles a procedure call, and may contain parameters passed to the task (e.g. START(Zip) or START(Laughter(30)). Note that starting a single PROCESS several times in a program creates a number of tasks executing the same copy of a PROCESS's code. The remaining parameters are optional. Task identifiers must be declared as variables of type processid. The stack size parameter consists of any integer-valued expression, and represents the number of words allocated for a stack space; if absent, START uses 200 as a default stack size. The priority parameter consists of an integer-valued expression; if it is not in the range 0..255, an execution error occurs. In the absence of a priority parameter, a started task inherits the main task's priority. Tasks terminate execution (by suspending themselves permanently) when they reach the end of their PROCESS code; however, their stack spaces occupy memory until the parent program finishes execution. The system prevents a program from terminating until all of its tasks have terminated.

NOTE - The description of tasks presented here is sufficient for describing the execution of programs containing processes. From the system's point of view, the entire system (of which user programs are merely a part) is called the main task; the other tasks (including system device drivers and user-defined processes) are known as subsidiary tasks. From the processor's point of view, there is no distinction between the main task and subsidiary tasks; at this level, they are functionally equivalent.

Processes are declared similarly to procedures; however, the reserved word PROCESS replaces the reserved word PROCEDURE. The syntax description presented below is derived from the formal syntax description for procedures in Appendix D of the Pascal User Manual and Report:

<process declaration> ::= <process heading> <block>

<process heading> ::=
    PROCESS <identifier> <formal parameter part> ;

RESTRICTIONS - PROCESSes must be declared in the outer (global) block of a program; they may not be declared within a PROCEDURE or another PROCESS. START may only be called from the main task; thus, processes cannot START other PROCESSes. Violation of this restriction causes an execution error.

WARNING - Tasks are not allocated their own heap space; all dynamic variables are allocated on the system heap. For this reason, using mark and release within a PROCESS is discouraged, as they may inadvertently remove variables created by other tasks. Sections of the operating system dealing with global resource management (e.g. file system and heap) are protected from contention between tasks; nevertheless, processes using these resources should do so carefully.

NOTE - Variable parameters passed to a PROCESS may require an associated semaphore in order to ensure mutually exclusive access to the actual parameter. Examples of PROCESS declarations:

```
PROCESS Zip;
begin
```

```
                    ...
                  end;

PROCESS BackgroundLaughter (Laughs: integer);
begin
  i := 0;
  if Laughs > 0 then
    repeat
      write('ha');
      i := (i + 1) mod Laughs;
      if i = 0 then writeln;
    until false;
end {BackgroundLaughter};
```

START assigns each task a unique value distinguishing it from other tasks. These values may be obtained by specifying a task identifier variable as a parameter to the START intrinsic; START assigns the value of the started task to the variable. Task identifier variables must be declared with the predefined type processid, and can be used in the same manner as pointer variables; i.e., the only valid operations are assignment and comparison with other task icentifier variables.

In the following example, two tasks are created with START; the variables PID1 and PID2 are assigned values identifying the tasks. Because these values are unique, this program writes "Truth" when executed:

```
Program a;

Var PID1,PID2: processid;

  PROCESS t;
  begin
    ...
  end;

begin
  START(t, PID1);
  START(t, PID2);
  if PID1 <> PID2 then writeln ('Truth');
end.
```

Each task is allocated an area of memory in which it can execute; because UCSD Pascal programs execute on stack-oriented machines, the memory area is called a stack space. Stack space is used to store parameters and variables, procedure call information, and segment procedure code. When a task exhausts its stack space, a "stack overflow" occurs, and the system must be restarted.

NOTE - The main task's stack space is coincident with the system stack, and is limited in size only by the amount of system memory available. Stack spaces for subsidiary tasks are created on the system heap by the START intrinsic; hence, PROCESS stack spaces are generally small compared to the main task's stack space.

WARNING - Because task stacks are allocated on the system heap, tasks are susceptible to destruction from inadvertent use of the mark

and release intrinsics. Never release below a started PROCESS.

NOTE - Stack sizes must be sufficient for the basic needs of a PROCESS; the minimum is: 4 words (procedure call info) plus the number of words used by local variables and parameters. Nested procedure calls take up similar amounts of space. Avoid calling segment procedures and/or procedures with large local data spaces (unless necessary -the utilities Disasm and Libmap can then be used to determine the size of the code and/or data segments involved and thus the amount of stack space required by the PROCESS.

NOTE - If the PROCESS body executed by a task is declared as a SEGMENT PROCESS, the code for the PROCESS body itself is read onto the task stack, and remains there while the task executes it; unfortunately, when the main program terminates, the system is not able to shut these tasks down in an orderly fashion; it must be rebooted. Therefore, segment processes should only be used in dedicated (i.e. nonterminating) programs.

Examples of stack size specification:

```
Program a;
Var PID: processid;
    I,J: integer;

  PROCESS p1;
  begin
    ...
  end;

begin
  I := 4;  J := 5;
  START(p1,PID);                 { stack space = 200 }
  START(p1,PID,10);              { stack space = 10  }
  START(p1,PID,(I + J)*100);     { stack space = 900 }
end.
```

Each task is assigned a task priority value between 0 and 255. A task's priority determines its ability to obtain the processor when other tasks are waiting to run. The processor's task scheduling policy is simple: no task may execute when a higher priority task is ready to run. This policy is enforced by ordering all tasks in the ready queue by their priority, and by performing a task switch when the task at the head of the ready queue has higher priority than the current task.

NOTE - When a task is inserted into the ready queue, it is placed behind all other tasks having priorities greater than or equal to its own.

NOTE - The main task's priority is 128. Priorities above 190 are reserved for the operating system. Starting a PROCESS with a priority higher than 128 immediately suspends the main task.

Examples of priority specification:

```
Program a;
```

```
Var PID: processid;
    I: integer;

  PROCESS p1;
  begin
    ...
  end;

begin
  I := 5;
  START(p1,PID,200);          { priority = 200 }
  START(p1,PID,200,90);       { priority = 90  }
  START(p1,PID,200,I*40);     { priority = 200 }
end.
```

Semaphores are declared as variables with the predefined type semaphore. Semaphores are used solely for task synchronization; they are shared by processes wishing to communicate with each other. Semaphores consist of two parts: a nonnegative integer counter and a queue for storing suspended tasks. However, semaphores are never accessed directly; they may be accessed only through predefined operators.

The principal semaphore operators are SEMINIT, WAIT, and SIGNAL.

SEMINIT initializes a semaphore variable by assigning it an initial count value and an empty wait queue.

WAIT checks the value of the semaphore count. If it is greater than zero, the count is decremented, and the current task continues to execute. Otherwise, the current task is suspended; it is placed in the semaphore's wait queue, and becomes a waiting task. The task at the head of the ready queue becomes the current task, and resumes its execution. Note that a task executing WAIT either continues as the current task or is suspended and becomes a waiting task.

SIGNAL examines the semaphore's wait queue. If it is empty, the semaphore's count is incremented. Otherwise, a waiting task is removed from the head of the wait queue; it is placed in the ready queue, and becomes a ready-to-run task. Note that a task executing SIGNAL either continues as the current task or becomes a ready-to-run task (because of a task switch).

NOTE - Tasks in a semaphore's wait queue are ordered by their priority.

Semaphores can be split into two classes (with respect to usage): binary semaphores and counting semaphores. Binary semaphores have only two states, as their counts only take on the values 0 and 1; they are used for mutual exclusion. Counting semaphores are so named because their count values can span the range of natural numbers; they are used for resource allocation.

WARNING - Semaphores must be initialized with SEMINIT before use; otherwise, system crashes occur. A semaphore's count value must not exceed 32767; otherwise, the count value wraps around to a negative value, leaving the semaphore in an undefined state.

When processes share a resource (usually a variable or an I/O

device), it is often necessary to protect the resource from being accessed by more than one PROCESS at a time; this is known as mutual exclusion. Mutual exclusion is ensured by placing all code which accesses the resource within a "critical section".

Critical sections are implemented with a binary semaphore by preceding the critical code with a call to WAIT and terminating the critical code with a call to SIGNAL. The binary semaphore is initialized to 1, indicating that the critical section is initially open; when a task is executing in a critical section, the semaphore count is guaranteed to be zero, ensuring that no other task may enter the critical section until it is available.

Example of mutually exclusive use of a console screen:

```
Program example;
Var Console: semaphore;

  procedure ConWrite(output: string);
  begin
    WAIT(Console);      { START critical section }

    { Critical section code }

    SIGNAL(Console);    { end critical section }
  end;

  PROCESS MsgWriter(WhoIAm: integer; MyMsg: string);
  begin
    repeat
      ConWrite(MyMsg)
    until false;
  end;

begin
  SEMINIT(Console,1);
  START(MsgWriter(1,'Shakespeare'));
  START(MsgWriter(2,'monkey'));
  START(MsgWriter(3,'typewriter'));
  .
  .
end.
```

Semaphores can be used to synchronize the execution of a group of processes so that each PROCESS's execution depends on the actions of another PROCESS. Processes used in this fashion are known as cooperating processes. Cooperating processes are implemented by assigning a private semaphore to each PROCESS. A PROCESS considers its own private semaphore to represent an event which must occur before it can resume execution; therefore, the PROCESS waits on its private "event" semaphore. Processes wishing to indicate the occurrence of an event do so by signalling the corresponding private semaphore, thus activating the suspended PROCESS which owns the private semaphore.

Cooperating processes and private semaphores are illustrated in the example below, which demonstrates buffered data transmission (concurrency speeds up this activity by allowing simultaneous filling and sending of different data buffers). The resources in need of management are the N data buffers shared by the processes FillBufs and

```
program clock;
const ClockVector =
{Vector for clock which interrupts 60 times/second};

var   tick: semaphore;
      pid: integer;

  PROCESS timer;
  const bell = 7;
        ThreeSeconds = 180 {ticks};
  var   ticks: integer;
  begin
    ticks := 0;
    repeat
      WAIT(tick);
      ticks := ticks + 1;
      if ticks >= ThreeSeconds then
        begin ticks := 0; write(chr(bell)); end;
    until false;
  end {timer};

begin {clock}
  SEMINIT(tick,0);
  ATTACH(tick,ClockVector);
  START(timer,pid,50,180);
end.
```

Time slicing refers to the allocation of processor time to each task in the ready queue. Time slicing does not alter the scheduling policy with respect to task priority; thus, it is relevant only to tasks having equal priority (with the proviso that higher priority tasks are not running on a regular basis).

The PDQ-3 system uses a system clock (which interrupts the processor every 60th of a second) and a high priority clock handler PROCESS to update the system time. Servicing clock interrupts has the side-effect of implementing processor time slices for user-defined tasks. The tasks appear to execute in parallel; actually, each task receives a period of processor time between clock interrupts, and then is tossed back into the ready queue until the other tasks have obtained their slice of the processor.

The circulation of tasks possessing equal priority is a consequence of the SIGNAL intrinsic and the priority of the clock handler PROCESS. A task switch always occurs when the system clock interrupts the processor; the current task is suspended and inserted behind all ready to run tasks having greater or equal priority. Thus, the ready queue serves as a first-in/first-out queue for processes with equal priority, while the regularity of clock interrupts ensures that the tasks recieve similar amounts of processor time.

The following example demonstrates time slicing; when executed, the program print final task counts that are approximately equal, indicating that the tasks receive similar amounts of processor time.

```
program RaceCondition;
const limit = 10000;
var   car1, car2, car3: integer;
```

SendBufs. FillBufs finds an empty buffer and fill it with data. SendBufs finds a full buffer and dispatches it. The private semaphores are BufAvail and BufFull. BufAvail indicates that a buffer has been sent and is available for filling; its initial value indicates that all buffers are initially available for filling. BufFull indicates that a buffer is full and available for transmission; its initial value reflects the lack of full buffers at the outset.

```
program Buffers;
const N = { number of available buffers };
var   BufFull, BufAvail: semaphore;

  PROCESS FillBufs;
  begin
    repeat
      WAIT(BufAvail);
      { ... Select an empty buffer and fill it ... }
      SIGNAL(BufFull);
    until false;
  end;

  PROCESS SendBufs;
  begin
    repeat
      WAIT(BufFull);
      { ... Select a full buffer and send it ... }
      SIGNAL(BufAvail);
    until false;
  end;

begin
  SEMINIT(BufFull,0);
  SEMINIT(BufAvail,N);
  START(FillBufs);
  START(SendBufs);
end.
```

The UCSD intrinsic ATTACH allows processes to be used as interrupt-driven device drivers. ATTACH assigns a machine-dependent hardware interrupt vector to a semaphore; from then on, the semaphore is signalled whenever the processor receives an interrupt through the indicated interrupt vector. NOTE - Interrupt vectors are described in the Hardware User's Manual.

WARNING - ATTACH treats semaphore arguments as permanent variables; therefore, semaphores attached to interrupt vectors must be declared in the outer block of either the main program or the appropriate device PROCESS. The machine knows only of the memory address of its semaphore argument, and continues to SIGNAL this address after every interrupt; it has no way of determining whether it is actually signalling a semaphore variable or merely damaging some unsuspecting code or data which happens to be allocated in space previously occupied by an attached semaphore. To wit, indiscreet use of ATTACH can adversely affect the system.

WARNING - Because the PDQ-3 system currently lacks a method for de-attaching semaphores, the system should be rebooted after running a user program containing attached semaphores.

This program uses the system clock to beep every 3 seconds:

```
    CheckeredFlag: boolean;

  PROCESS racer(var counter: integer);
  begin
    counter := 0;
    repeat
      counter := counter + 1;
      if not CheckeredFlag then
        CheckeredFlag := counter >= limit;
    until CheckeredFlag;
    writeln(counter);
  end;

begin
  CheckeredFlag := false;
  START(racer(car1));
  START(racer(car2));
  START(racer(car3));
end.
```

Tasks can delay their execution for an integral number of clock ticks by calling a special form of the unitread intrinsic. The system maintains delayed tasks in a special task queue known as the time queue.

Example of time delay:

```
program Somnambulator;
type ticks = integer;
var  pid: processid;

  procedure sleep(time: ticks);
  const timeQ = 0;
        dummy2 = 0;
  var   dummy1: array [0..5] of integer;
  begin
    unitread(timeQ,dummy1,dummy2,time);
  end;

  PROCESS snooze;
  begin
    repeat
      writeln('Snore!');
      sleep(5);
    until false;
  end;

  PROCESS catnap;
  begin
    repeat
      writeln('yawn...');
      sleep(2);
    until false;
  end;

  PROCESS hyper;
  begin
    repeat
      writeln('WAKEUP!!!');
```

```
    until false;
  end;

begin
  START(snooze,pid,100,100);
  START(catnap,pid,100,100);
  START(hyper ,pid,100,100);
end.
```

6.7 Multi-terminal Applications

This section describes the role of UCSD Pascal's concurrency features and the PDQ-3 asynchronous I/O system in the development of multi-terminal programs. A multi-terminal hardware configuration consists of a PDQ-3 communicating with two or more terminals over serial lines. A multi-terminal program assigns a task to each of the terminals; each task is responsible for receiving input from its assigned terminal, initiating actions specified by the terminal user, and sending output back to the terminal. When a task initiates an I/O operation, it is suspended until the I/O operation is completed; this allows other tasks to proceed with their execution.

The variables declared in a multi-terminal program can be divided into two classes: private variables, and shared variables. Private variables define the state of each user. They are declared in the terminal process; thus, each task is allocated its own set of private variables. A typical example of a private variable is the file variable which each task uses for reading and writing to its own terminal. Shared variables define the resources being shared by all tasks. They are declared in the main program; because they are subject to task contention, they must be guarded by semaphores. A typical example of a shared variable is a disk file which is accessible to every task.

The example presented below illustrates the adaptation of a single-user program to a multi-terminal environment. Most of the conversion effort involves the determination of which global variables in the program should be shared, and which should be private. (Note that non-global variables are unaffected by the conversion.) Semaphores are then assigned to each shared variable; each reference to a shared variable must be guarded by its semaphore to ensure mutual exclusion of the terminal tasks. A terminal process must be written; it contains the declarations of all private variables, and does nothing more than initialize its private variables and call the main program routine (i.e. the procedure responsible for starting the entire program).

The main program routine must be modified so that references to private-cum-global variables become formal parameter references. Each terminal task passes the relevant private variables to the parameterized version of the main program routine.

NOTE - See section 3.1.0 for restrictions imposed by interaction between tasks and segments.

NOTE - In addition to the changes mentioned above, it is necessary to locate all critical sections (see section 3.0.1.0) and protect them with semaphores. Failure to identify and protect all potential critical sections can cause multi-terminal programs to fail intermittently.

Page 153 omitted

Page 152 had Page 154 printed on the back

Double sided printing

Example of multi-terminal program:

```
program CrossTalk;
var LogLock : semaphore;
    LogFile: text;                    { shared }
    pid: processid;

  procedure LogChar (Ch : char; Receiver : string);
  begin
    wait (LogLock);
    writeln (LogFile, 'Received ', Ch, ' from ', Receiver);
    signal (LogLock);
  end;

  process ATerminal (InName, OutName : string);
  const EscapeChar = ':';
  var Ch : char;                      { private }
      InFile : interactive;
      OutFile : text;
  begin
    reset (InFile, InName);
    rewrite (OutFile, OutName);
    repeat
      read (InFile, Ch);
      write (OutFile, Ch);
      LogChar (Ch, InName);
    until Ch = EscapeChar;
  end {ATerminal};

begin
  { init tasks & shared variables }
  seminit (LogLock, 1);
  rewrite (LogFile, 'CONSOLE:');
  start (ATerminal ('REMIN1:', 'REMOUT2:'), pid, 500);
  start (ATerminal ('REMIN2:', 'REMOUT1:'), pid, 500);
end.
```

Advanced Computer Design

A Subsidiary of Computer Services Group, Inc. / 7584 Trade St. / San Diego, CA 92121 / (714) 578-9620

ADVANCED SYSTEM EDITOR

****

The standard UCSD PASCAL Editor is designed to help the user create new text files and update old ones. Some of the features include text insertion and deletion, exchanging existing characters with new ones, finding character patterns, and replacing them with new patterns. Moving throughout a file is accomplished by J(umping to either the beginning or the end of the file, or to a marker. More precise movement may be controlled through the use of the Space, Backspace, Return, or Vector keys.

The following list briefly describes some of the additional features that ara available with the optional Advanced System Editor. This improved version of the standard editor is intended to be used by both experienced users of the standard editor, who wish to greatly facilitate file maintainence, and by inexperienced users, who simply want the convenience offered by the Advanced System Editor.

LARGE FILE HANDLING:

Whereas the standard editor can only edit files that will fit into the computer's internal memory, the Advanced System Editor (ASE) is capable of editing arbitrarily large text files, limited only by on-line disk capacity.

EXTENDED EXCHANGE:

The X(change command now allows the user access to the vector keys for moving around the screen, to insert and delete characters, and to type in any direction. For example, while in the X(change mode, the user can move up 10 lines, delete a character, specify a typing direction of "down", and by typing 10 stars, draw a vertical line of 10 stars. This is quite an improvement over the standard X(change mode where exchanges take place in a character for character fashion, from right to left only.

RECURSIVE EDIT:

The new E(dit command allows the user, in the middle of editing one file, to begin editing another. The number of files that may be pending in the E(dit mode is virtually unlimited. Thus the user can, for instance, inspect and set markers in an existing file before copying part of it into his workspace. This same operation in the standard editor requires several keystrokes, several disk accesses, and several unnecessary minutes.

F(ind and R(eplace:

These two commands have been enhanced to include "wild-card" tokens (e.g. "the $ tree" matches any three word sequence beginning with "the" and ending with "tree"), case insensitive matching ("ThE" matches "tHe"; though case-sensitive matching is still available), and user-controlled retention of patterns, which enables the user to maintain several substitution and target strings at the same time.

"Special" Keys:

Five special keys that behave like macro invocations are recognized by the ASE. These keys may be defined to be any sequence of keystrokes (e.g. a key may be defined to be "F/begin/". Each time the key is typed, the cursor will move to the next "begin".).

Keystroke Optimization:

In order to reduce the number of keystrokes needed to perform simple movements within a file, four new commands have been added. O(pposite displays the previous page of text (" P(age" must be typed to do the same thing with the standard editor). B(eginLine and L(ineEnd move the cursor to the beginning and end, respectively, of the current line (this was formerly accomplished only by backspacing or spacing, respectively). Finally, G(etchar, followed by a character, moves the cursor to the next occurance of that character.

Better Terminal Handling:

The ASE handles "prefixed" terminals (e.g. DEC VT-52) in an improved fashion. In addition, the ASE can take advantage of the Insert Character, Delete Character, and Reverse Scrolling features of many terminals.

For more information on the Advanced System Editor, write to Advanced Computer Design, 7584 Trade St., San Diego, California 92121.

Appendix A was not included

APPENDIX B

PRICE LIST
****

	MODEL NUMBER	DESCRIPTION	PRICE	DISCOUNT SCHEDULE
°	PDQ-3S-X-YYY-Z	PDQ-3 COMPUTER SYSTEMS		
	PDQ-3S-4-064-1	PDQ-3 Computer System Including: PDQ-3 PASCAL CPU, Real Time Clock, Interval Timer, CRT Console and Printer Interface, DMA Floppy Disk Controller, 64KB of Memory, 4 Quad Size Slot Q-Bus Backplane, Power Supply and two (2) 8" Single Sided Floppy Disk Drives, Desk Top or Rack Mount Chassis.	$6,995.	B
	PDQ-3S-8-YYY-Z	8 Quad Size Slot Backplane and Power Supply	Add: $ 950.	
	PDQ-3S-X-128-Z	64KB of Add on Memory	Add: $ 985.	C
	PDQ-3S-X-YYY-2	Double Sided Floppy Drives	Add: $ 595.	C
	PDQ-3CPU	PDQ-3 PASCAL CPU Module with CPU, Multiplexed CRT Console and Printer Interface, Real Time Clock, Interval Counter, Interrupt Logic, DMA Logic, DMA Floppy Disk Controller	$3,195.	A
°	PDQ-3PPL	Double Density Module	Add: $ 180.	
°	PDQ-3CHS-X-Z	SYSTEM CHASSIS		
	PDQ-3CHS-4-1	System Chassis with 4 Quad Size Slot Q-Bus Backplane, Power Supply and two (2) Single Sided 8" Floppy Drives.	$2,995.	C
	PDQ-3-10MB WINC	10MB Winchester Fixed Disc Drive LSI-11/Smart Interface	$4,595.	C
	PDQ-3-30MB WINC	30MB Winchester Fixed Disc Drive LSI-11/Smart Interface	$5,895.	C
	PDQ-3-20MB STRM	20MB Cartridge Backup Streamer	Add: $1,795.	C
	Z19	(VT-52 Compatible) Terminal	$ 895.	B

APPENDIX C

DISCOUNT SCHEDULE
****

Schedule "A"

Quantity:	5-9	10-24	25-49	50-99	100+
Discount:	10%	15%	20%	30%	35%

Schedule "B"

Quantity:	5-9	10-24	25-49	50-74	75-99	100+
Discount:	10%	15%	20%	25%	27.5%	30%

Schedule "C"

Quantity:	10-24	25-49	50-99	100+
Discount:	4%	8%	12%	16%